

DRAFT User's Manual for Groundwater/Surface Water Interaction Using ADH

Part of the CHSSI project on High Fidelity Simulation of Littoral Environments

S. E. Howington, R. C. Berger, E. W. Jenkins,
J. P. Hallberg, K. R. Kavanagh, and J. H. Schmidt ¹

July 8, 2003

¹USAE Waterways Experiment Station and North Carolina State University

Contents

1	Introduction	1
1.1	Scientific Problem	1
1.2	Technical Approach	1
1.3	Brief Model Description	2
1.4	Mesh Refinement/Coarsening	3
1.5	Parallelization	4
1.6	Matrix Preconditioners	4
1.7	Application	6
1.8	Additional Benefits	6
1.9	Status and Plans	6
2	Model Description	8
3	Equations	11
3.1	Richards Equation	11
3.2	Shallow Water Equations	12
4	Solvers and Preconditioners	14
4.1	Newton-Krylov-Schwarz methods	14
4.1.1	Nonlinear Solvers	14
4.1.2	The Linear Iteration and Preconditioning	16
4.1.3	Schwarz Preconditioners	16
4.1.4	Parallel implementation	19
5	Model Input	20
5.1	Boundary Condition Files	20
5.1.1	Operation Parameters	22
5.1.2	Iteration Parameters	23
5.1.3	Materials	23
5.1.4	Boundary Strings	30
5.1.5	Solution Controls	31
5.1.6	Time Controls	32
5.1.7	Output Control	32
5.2	Hot Start File	32

6	Using the GMS with ADH	34
6.1	Problem Solving with ADH	34
6.1.1	Traveling Plume	35
6.1.2	Running pre_ADH	54
6.1.3	Running ADH	54
6.2	References	54
A	Constructing Model Files	57
A.1	3-D Mesh Files	57
A.2	Boundary Conditions	57
A.3	Hot Start File	62

List of Tables

3.1	Typical van Genuchten parameters for the soil textural groups used[?].	12
5.1	Cards	21
5.2	van Genutchen Parameters	26
A.1	Operation Parameter Cards	58
A.3	Material Property Cards	59
A.2	Iteration Parameter Cards	61
A.4	Material Meshing Control Cards	62
A.5	String Structures	63
A.6	Initial and Boundary Conditions	64
A.7	Initial and Boundary Conditions (continued)	65
A.8	Solution Control Cards	66
A.9	ADH input file template.	67

List of Figures

1.1	Typical interaction of groundwater and surface water.	2
1.2	Schematic showing the refinement and coarsening of a single tetrahedral element, adding or removing one element and one node.	3
1.3	An example fine mesh with element boundaries shown in black and four preconditioner subdomains. The light gray elements are the overlapping regions.	4
1.4	Schematic showing the mesh partitioning scheme. On the left, a simple mesh is divided among three processors. Nodes are assigned to processors. Elements may be shared by processors. On the right is the view from processor 0 showing interior nodes, border nodes, and ghost nodes.	5
1.5	The coarse-mesh basis function is produced by summing all the fine-mesh basis functions within the subdomain. The resulting coarse-mesh basis function is constant except in the elements shared with the other subdomains.	5
1.6	The initial Poplar Creek watershed computational mesh.	6
2.1	Rectangular Block of Soil.	9
2.2	Transported Solute Concentration.	10
3.1	The ψ - S (left) and ψ - k_r (right) models used in the column drainage test.	12
4.1	Subdomain Splits in One-Dimension for Two Subdomains	16
4.2	Coarse Grid Basis Function in One-Dimension	18
5.1	Suction-Saturation Relationship for Typical Sand, Silt, and Clay	27
5.2	Relative Conductivity for Sand, Silt, and Clay	28
6.1	Initial shape and location of solute.	35
6.2	Completed 2-D Mesh	36
6.3	Selection of the Boundary Nodes for Face 1	40
6.4	Selection of elements of solute plume.	51
6.5	Initial Plume	53

6.6	Movement of a solute plume without dispersion or diffusion. . . .	55
-----	---	----

Abstract

Guidelines are presented for using the US Army Corp of Engineers (USACE) ADH modeling software to model exchange of fluids between surface and subsurface systems. ADH is used in conjunction with the Department of Defense Groundwater Modeling System (GMS). This manual includes both reference material and simple tutorials for the ADH software. The examples herein demonstrate the basic capabilities of ADH and the interface between ADH and the GMS.

Chapter 1

Introduction

1.1 Scientific Problem

The Department of Defense (DoD) must assess the environmental impact of its activities at both present and formerly used facilities. When warranted, the DoD must enact remedial measures to address environmental problems. The potential costs associated with environmental remediation at DoD sites is staggering. In addition to the cost of remediation, the DoD risks reduced or prohibited access to its training facilities unless environmental concerns are addressed adequately. For these reasons, accurate environmental assessments and effective remedial designs are essential.

Thorough environmental assessment requires that the ecosystem be examined in a more holistic fashion than is customary. Traditionally, each part of a hydrologic system (groundwater, wetland, drainage basin, etc.) has been modeled individually, treating the other parts of the hydrologic system as sources or sinks that are assumed to be constant or described with simple empirical functions. Often, the disparate temporal or spatial scales among these systems justify this uncoupled treatment. However, in some cases, the degree of interaction or the uncertainty in the magnitude of sources or sinks requires that multiple components of the hydrologic system be considered simultaneously. Examples include groundwater-driven hydrology in wetlands, contaminant exchange between surface and subsurface systems, and heterogeneous, transient infiltration. In many locations, such as south Florida, the groundwater and surface water systems are so tightly coupled that they are virtually inseparable. Figure 1 is a schematic showing several typical points of interaction between groundwater and surface water systems.

1.2 Technical Approach

The ADH (ADaptive Hydrology) groundwater model, constructed at the Engineering Research Development Center (ERDC), is being extended for use as

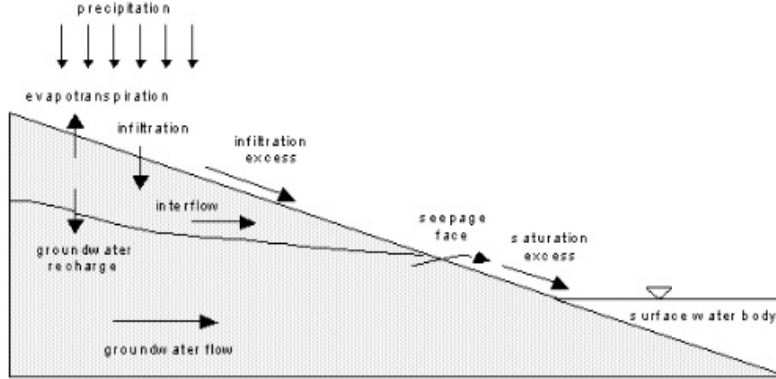


Figure 1.1: Typical interaction of groundwater and surface water.

the model foundation. Knowledge gained through the creation of the HIVE2D surface-water model, also at the ERDC, is helping develop a new surface water modeling capability within the ADH framework.

ADH couples three-dimensional, unsaturated groundwater modeling to two-dimensional, shallow water modeling at the surface. Previous attempts to account for the interaction of groundwater and surface water have linked a saturated groundwater flow model with an overland flow model through a one-dimensional (vertical) representation of the unsaturated zone. Such a one-dimensional unsaturated flow interface typically uses a simplified depiction of the unsaturated flow as wet-dry interface propagation or, even simpler, as a homogenized-volume represented by a constitutive equation. The ADH model is a technical advance because it maintains full three-dimensionality in the unsaturated zone, permitting the simulation of perched aquifers, inter-formation flow (lateral flow in the unsaturated zone), and infiltration processes in heterogeneous systems. The drawback in taking this approach is the large computational effort required.

1.3 Brief Model Description

ADH approximates the solution to the Richards equation for groundwater flow and either the diffusive wave equation or the full shallow water equations for surface water flow. The Richards equation is a combined water balance and momentum equation for saturated and partially saturated soil. This equation is

Figure 1.2: Schematic showing the refinement and coarsening of a single tetrahedral element, adding or removing one element and one node.

non-linear because some of the coefficients (saturation and relative permeability) depend on the unknown head. The diffusive wave equation for overland flow arises by neglecting the acceleration terms in the full St. Venant equations (for example, Singh, 1996).

Finite elements are used to discretize the domain. The approximation is piecewise linear in space and piecewise constant in time. Groundwater flow is solved in three dimensions using tetrahedra. The diffusive wave equation is approximated on triangles that comprise a surface of the three-dimensional groundwater flow mesh. Nodes located on the overland flow face are dual valued, with an overland flow head and a groundwater head. The two flow regimes communicate through boundary fluxes computed at the surface of the groundwater system. By communicating only through fluxes, the model avoids a problem found in other models coupling groundwater and surface water. Many of these models switch boundary conditions, using Dirichlet (head) boundaries when the depth in the surface water is non-zero, and Neumann (flux) boundaries for recharge when the surface water has zero depth. Presently, the two flow regimes update their fluxes only at each time step. As model testing proceeds, it may be necessary to enforce this communication at each non-linear iteration.

1.4 Mesh Refinement/Coarsening

Many of the physical problems to be addressed with the ADH model contain steep and moving spatial gradients in the solution variables. Examples of these gradients include a moving saturation front or intermittent well in the groundwater system, a traveling wave in the surface water system, or a contamination front in either system, for example. Capturing these phenomena with a fixed-mesh model would require extremely fine mesh resolution throughout the domain. Such resolution is not practical for many problems and is not efficient use of resources for most problems. For these reasons, the ADH model uses local mesh refinement and coarsening to add and delete resolution, as needed, to capture steep gradients.

Splitting or merging elements is based on an explicit error indicator. Presently, the model uses an inexpensive, gradient-based indicator, but more accurate (and costly) indicators (Schmidt, 1997) are available in the model. Elements slated for refinement are divided using the edge bisection scheme by Liu and Joe (1995) (Figure 2). Edges are ranked by the refinement level of its nodes and by their length. The 'oldest' edge in an element is split first. If all elements are the same 'age', the longest edge is split first.

Figure 1.3: An example fine mesh with element boundaries shown in black and four preconditioner subdomains. The light gray elements are the overlapping regions.

1.5 Parallelization

ADH has been constructed to take advantage of parallel computer architectures. The domain is subdivided in a data parallel scheme shown by a simple example in Figure 3. Nodes are distributed to processors uniquely. Elements that fall on processor boundaries are shared. Ghost nodes are created for those nodes that reside off processor, but contribute to a shared element. Border nodes are on-processor, but are seen as ghost nodes by another processor. Thus, border nodes must communicate information with other processors.

Processor partitions generally will contain elements that are spatially adjacent to each other because this tends to minimize inter-processor communication. Therefore, as the mesh is refined locally near a large gradient, a majority of the additional elements and nodes can be created on only a few processors. Thus, local mesh adaption creates an inherent workload imbalance among the processors. Periodic repartitioning is needed to maintain the load balance for the dynamic system. Mesh refinement occurring on one processor must be communicated to other affected processors.

Inter-processor communication is handled with the standard Message Passing Interface (MPI) libraries to ensure the model's portability among many HPC machines. Thus far, the model has been run on the Cray T3E, IBM SP, and SGI Origin 2000 at the Major Shared Resource Center in Vicksburg.

1.6 Matrix Preconditioners

Recent research indicates that for many problems, including groundwater transport in naturally heterogeneous soils, significant resolution is necessary to produce qualitatively correct answers (Tompson and Gelhar, 1990, Howington et al, 1997). This revelation comes as the trend in physical problem dimension and complexity continues to increase rapidly. In tandem, the trend toward larger physical dimension and finer resolution is leading to enormous increases in the number of nodes and elements in a typical simulation. The ADH model is implicit in time, requiring the simultaneous solution of large non-linear algebraic systems. The number of nodes and the degrees of freedom per node determined the size of this system of equations. An inexact Newton's method is used to linearize the problem. Thus, a linear system must be solved for each Newton iteration and several Newton iterations may be required for each time step.

A general concern exists with the solution of these large linear systems. The number of iterations required to solve the system grows with problem size for most schemes. Therefore, the work required for matrix assembly, etc. increases linearly with the number of processors, but the work required to solve the system

Figure 1.4: Schematic showing the mesh partitioning scheme. On the left, a simple mesh is divided among three processors. Nodes are assigned to processors. Elements may be shared by processors. On the right is the view from processor 0 showing interior nodes, border nodes, and ghost nodes.

Figure 1.5: The coarse-mesh basis function is produced by summing all the fine-mesh basis functions within the subdomain. The resulting coarse-mesh basis function is constant except in the elements shared with the other subdomains.

can grow more rapidly. With clever preconditioning of the linear system, this growth in the number of iterations can be dramatically reduced (Tompson et al, 1994).

A domain-decomposition approach was chosen to precondition the linear system because these are well suited for parallel implementation. When these subdomains are overlapping, these are known as Schwarz preconditioners. The preconditioner options in ADH are:

- Point Jacobi
- One-level Additive Schwarz
- Two-level Additive Schwarz
- Two-level Hybrid Schwarz

Point Jacobi preconditioning makes no use of domain decomposition. The remaining preconditioners divide the domain into overlapping subdomains. Figure 4 shows a sample fine mesh and four subdomains.

One-level additive Schwarz is simply a block Jacobi preconditioner. A fine-mesh solve on each subdomain is followed by an interpolation back to the full preconditioning matrix. To extract each subdomain from the larger system, one must assume boundary conditions. Zero Dirichlet boundary conditions are used on the subdomain boundaries. Two-level additive Schwarz schemes add a full-domain coarse mesh solve to the subdomain solves. Basis functions for the fine mesh elements are summed to create a single basis function for that subdomain (Jenkins et al, in preparation). This summed basis function for one of the four subdomains in the sample mesh is shown in Figure 5. The coarse problem consisting of a single matrix entry per subdomain is solved. Parallelizing the coarse mesh is not yet required because each processor can perform the coarse solve independently. The two-level additive and two-level hybrid schemes combine the fine and coarse mesh information differently. This domain decomposition approach is, effectively, a simple, multigrid preconditioner on an unstructured mesh, without the complexity of creating, maintaining, and parallelizing multiple, nested, meshes.

Figure 1.6: The initial Poplar Creek watershed computational mesh.

1.7 Application

The ADH model has been applied to several example problems and application to a field site is underway. Among the example problems are drainage through a heterogeneous soil column and rainfall/runoff in a simple test basin. The column problem is intended to demonstrate the model's capabilities in simulating drainage in heterogeneous soil. The simple test basin is being used to explore the fluxes across the ground surface and evaluate the performance of the overland flow model.

A field problem has been constructed to test the model. The Poplar Creek drainage basin at Camp Shelby near Hattiesburg, MS has been studied extensively. The numerical mesh to be used in the initial simulations for this field site is shown in Figure 6. Rain will be applied to the surface and flux in the creek will be compared to measured discharge.

1.8 Additional Benefits

This work was motivated by the need to simulate the interaction between groundwater and surface water. However, by adopting a strategy that attaches problem-specific routines (groundwater flow equations and overland flow equations) to a single, computational engine, several software development and maintenance advantages have become apparent. The computational engine contains matrix solvers, mesh adaption routines, generic finite element routines, and parallel communication routines. These difficult and time-consuming parts of the code remain virtually unaffected by many changes to the groundwater or surface water routines.

Another major issue with a development of this magnitude is code maintenance. If written in a modular fashion, a single code may perform simulations for groundwater, surface water, or coupled systems with little overhead penalty. Therefore, this code may circumvent the need to maintain individual codes, which often contain similar modules. By sharing related modules, advancing capabilities are kept in step for each of the potential applications. Our hope for the future is to extend the model to include other components of the hydrologic system. The difficulty lies in keeping the code at a manageable size and in keeping the components sufficiently modular to permit enhancements and maintenance.

1.9 Status and Plans

The ADH model presently solves coupled groundwater and surface water flow using a diffusive wave approximation for overland flow. The model is being

tested against simple problems for which analytical approximations are possible and against data from instrumented watersheds. Plans exist to upgrade the surface water model to solve the full shallow water equations. A method to handle canals efficiently in one dimension will be explored. A constituent flux will be added at the boundary between the flow regimes to accommodate constituent transport between surface and subsurface waters.

Because the mesh partitioning among subdomains (not simply among processors) defines the coarse mesh for the Schwarz preconditioner, the partitioning scheme deserves careful study. Preliminary simulations indicate that the convergence rate is sensitive to the shape of these subdomains. Likewise, there is a delicate balance when choosing the number of subdomains. Having too many subdomains creates a very large coarse mesh problem, while too few subdomains requires the solution of large problems for each subdomain.

Chapter 2

Model Description

The interaction between these two physical regimens and therefore different representative equations, is communicated between the two models by exchange of fluxes. The sw-gw flux appears in the surface water (diffusive wave) model as a source or sink for water. In the groundwater this flux is implemented as a result of the weak form of the equations and appears as a flux through the boundary surface. The actual magnitude of the sw-gw flux uses only the groundwater heads to calculate the saturation of the ground. The relative conductivity uses the surface water heads on the sw-gw interface and the groundwater head below the interface.

The diffusive wave model and the groundwater models are loosely linked. That is, the overland flow is calculated first using previously defined groundwater heads. The groundwater calculations now use these calculated overland flow variables. This is repeated each time step.

Overland flow model

The overland flow model includes rain and groundwater interaction as sources and sinks. The surface flow is modeled using the diffusive wave formulation. This is a simplification of the shallow water equations in which the acceleration terms are removed. The overland flow depths are interpolated linearly. The finite element formulation uses Galerkin Least Squares (GLS) for stabilization.

The ADH Model is a software package used to describe both saturated and unsaturated groundwater, overland flow, and 3D Navier-Stokes problems. This particular manual contains descriptions for the Navier-Stokes solutions. The model was designed to work in conjunction with the DoD Groundwater Modeling System (GMS). The GMS is a modeling package for building models, running simulations, and visualizing results. For further information regarding the GMS, contact the USACE Waterways Experiment Station or visit the website at <http://chlnet.wes.army.mil/software/>.

Water quality problems are commonly examined with groundwater modeling systems. It is often important to view the movement of a solute through soil media due to the action of a natural groundwater flow or forced pumping. Field studies are often very complex with complicated topography and soil structure

interactions. However, the basic mechanics of the problem can be modeled quite simply. ADH calculates piezometric heads for a flowfield and further derives the velocity field. From these calculations, transport equations are solved for the concentrations of specified solutes.

As an example, imagine a horizontal “column” of saturated soil as shown in part *a* of Fig. ?? . Lowering the piezometric head across the length of the column would result in a simple groundwater flow through the column. If a solute at a specified concentration was located at the upstream end, the solute would migrate in the direction of the groundwater flow. Part *b* of Fig. ?? shows the resulting solute concentration after the migration of the solute through the media. Although simplistic in nature, this example represents the case where a solute is washed across its boundary into the surrounding “clean” soil due to the movement of groundwater. This model clearly demonstrates the capabilities of ADH.

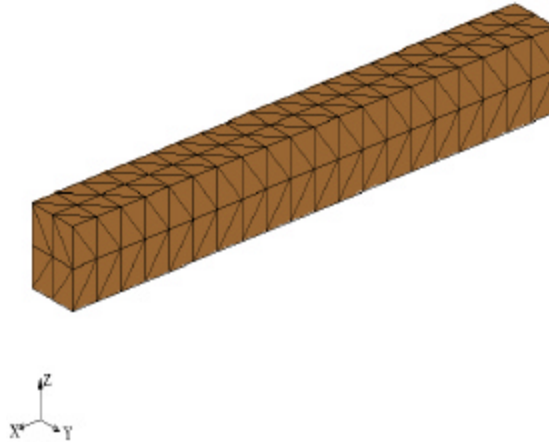


Figure 2.1: Rectangular Block of Soil.

Only three files are needed to run a model in ADH. These files are the mesh file, the boundary conditions file, and the hot start file. The mesh file must be constructed first and can be generated directly with the GMS. Once a mesh file has been constructed, the boundary conditions for the problem and operating parameters for ADH must be specified in the boundary conditions file. Many of the operational parameters and boundary conditions can be specified through the use of the GMS, using it’s tools to select boundary nodes and faces as well

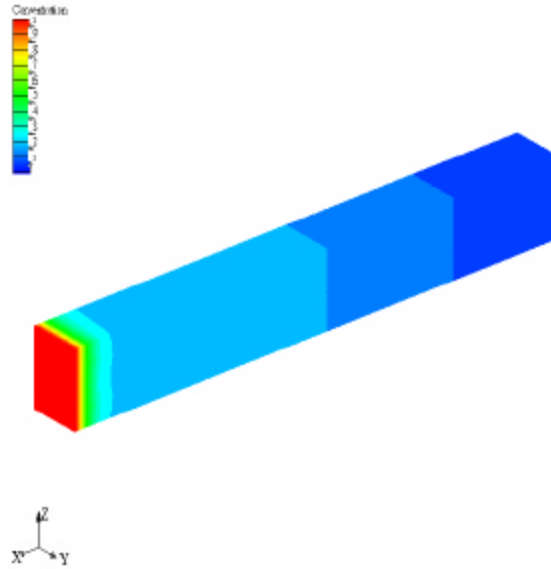


Figure 2.2: Transported Solute Concentration.

as entire sides of a domain. The hot start file is then generated to establish the initial conditions of the problem.

Once the three required files have been created, the ADH model is run with the command

`adh filename`

where *filename* is the root of the model's filenames, i.e. for a model named `layer` the following three files would be required `layer.3dm`, `layer.hot` and `layer.bc`. All three files must have the same *filename* as their root followed by one of three suffixes. After the model is run, the GMS can be used to visualize the results. Visualization capabilities of the GMS include contours of the flow head and chemical concentrations. With the GMS it is also possible to cut cross sections and overlay the mesh, allowing the user to view things such as grid dependencies in the solutions.

Chapter 3

Equations

3.1 Richards Equation

The mixed form of Richards' Equation is [?]

$$S_S S(\psi) \frac{\partial \psi}{\partial t} + \eta \frac{\partial S(\psi)}{\partial t} = \nabla \cdot [K_S k_r(\psi) \nabla (\psi + z)] + W \quad (3.1)$$

where ψ is pressure head; S_S is the specific storage, which accounts for water compressibility and aquifer elasticity; $S(\psi)$ is the water saturation or volumetric fraction of pore space occupied by water; η is the porosity or volumetric void fraction; K_S is the water-saturated hydraulic conductivity; k_r is the relative permeability of the media; and W is a source/sink term.. In this formulation, the z axis is the vertical direction oriented positively upward. Both S and k_r are functions of ψ . K_S and S_S are provided as data. This form has been shown to be mass-conserving when low-order time integration is used [?].

S is defined by

$$S = S_r + \frac{(1 - S_r)}{[1 + (\alpha\psi)^n]^m} \quad (3.2)$$

where S_r is the residual saturation, and α , m , and n are parameters specific to the fluid and soil. Van Genuchten found good agreement with experimental data for $m = 1 - 1/n$. With the water retention model from van Genuchten, Mualem's relative permeability function is

$$k_r = \frac{[1 - (\alpha\psi)^{n-1} [1 + (\alpha\psi)^n]^{-m}]^2}{[1 + (\alpha\psi)^n]^{m/2}} \quad (3.3)$$

where again $m = 1 - 1/n$.

The Environmental Protection Agency [?] (referring to work by Carsel and Parrish [?]) suggests typical parameters for general soil textural groups. For

san, silt, and clay soils like those in the test problem, typical parameters are given in Table 3.1 and these functions are plotted in Figure 3.1. Under static conditions, capillary pressure head corresponds to height above the water table (line of atmospheric pressure). Therefore, the $\psi - S$ relationship shows the expected saturation profile above the water table. Note the strong dependence of relative permeability on capillary pressure head which makes solution of Richards' equation challenging.

Table 3.1: Typical van Genuchten parameters for the soil textural groups used[?].

Soil Type	S_r (-)	α (1/ft)	n (-)
Sand	0.105	4.420	2.68
Silt	0.074	0.487	1.37
Clay	0.179	0.244	1.09

Figure 3.1: The ψ - S (left) and ψ - k_r (right) models used in the column drainage test.

The cost of evaluating the nonlinearities in (3.2) and (3.3) on a three dimensional mesh is prohibitive [?, ?] and it is far more efficient to approximate S with a spline. We used a piecewise linear splines in the computations reported here.

Implicit temporal integration in three space dimensions leads to large nonlinear equations at each time step. Newton-Krylov-Schwarz methods solve nonlinear equations by using Newton's method with a Schwarz domain decomposition preconditioned Krylov method to approximate the Newton step. These methods have been used in computational fluid dynamics for some time [?, ?, ?]. In § 4.1 we describe the nonlinear and linear solver issues raised by Richards' equation, introduce some notation from [?] for domain decomposition, and use that notation to describe our approach to Schwarz preconditioning.

3.2 Shallow Water Equations

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + H = 0 \quad (3.4)$$

where

$$U = \begin{pmatrix} h \\ uh \\ vh \end{pmatrix} \quad (3.5)$$

$$F = \begin{pmatrix} uh \\ u^2h + \frac{1}{2}gh^2 - \frac{h\sigma_{xx}}{\rho} \\ uvh - \frac{h\sigma_{yx}}{\rho} \end{pmatrix} \quad (3.6)$$

$$G = \begin{pmatrix} vh \\ uvh - \frac{h\sigma_{xy}}{\rho} \\ v^2h + \frac{1}{2}gh^2 - \frac{h\sigma_{yy}}{\rho} \end{pmatrix} \quad (3.7)$$

$$H = \begin{pmatrix} 0 \\ gh \frac{\partial z_h}{\partial x} + n^2 g \frac{u\sqrt{u^2+v^2}}{h^{1/3}} \\ gh \frac{\partial z_h}{\partial y} + n^2 g \frac{v\sqrt{u^2+v^2}}{h^{1/3}} \end{pmatrix} \quad (3.8)$$

Chapter 4

Solvers and Preconditioners

In this paper we discuss the design and implementation of a Newton-Krylov-Schwarz solver for the implicit temporal integration on an unstructured three-dimensional spatial mesh of Richards' equation for groundwater flow in the unsaturated zone. This work is one of the first to use two-level additive Schwarz preconditioners in groundwater simulations and uses aggregation ideas from the algebraic multigrid literature [?] to construct a coarse mesh for two-level additive Schwarz methods. Our coarse mesh differs from other approaches, for example that in [?], in that no coarse mesh geometry need be created and we do not need geometric information about the subdomains. A different approach to this problem, based on multigrid methods, has been described in [?, ?, ?, ?]. Our preconditioner has also been applied to surface water flow [?].

4.1 Newton-Krylov-Schwarz methods

The weak formulation of Richards' equation leads to finite element discretizations that are implicit in time. An elliptic partial differential equation must be solved at each time step. After discretization one obtains a large system of nonlinear equations. In this section we begin with a description of inexact Newton methods [?] in general terms and then show how Newton-Krylov-Schwarz methods fit in that framework.

4.1.1 Nonlinear Solvers

We express nonlinear equations in the general form:

$$F(\xi) = 0. \tag{4.1}$$

We will assume that a solution ξ^* exists and that a good approximation to ξ^* is available. This latter assumption is appropriate in the context of implicit temporal integration, where ξ^* is the solution at the new time step and the initial approximation (the predictor) is an interpolation of solutions at previous

times. We let $F'(\xi)$ denote the Jacobian of F at a vector ξ . We assume that $F'(\xi)$ is Lipschitz continuous in ξ and $F'(\xi^*)$ is nonsingular. These are standard assumptions in nonlinear equations [?] and are valid for the differential equations under consideration in this paper. The smoothness assumptions can be violated by the nonlinearities for Richards' equation in some cases [?], but this nonsmoothness can be managed by approximation with a well-chosen spline. Even the nonsmooth effects of a piecewise linear spline, which we use in our implementation, are benign.

We will describe Newton's method for (4.1) in the standard way [?, ?] in terms of the transition from a current approximation ξ_c to the solution ξ^* to a new approximation ξ_+ and describe the convergence in terms of the relative sizes of the new error $e_+ = \xi_+ - \xi^*$ and the current error $e_c = \xi_c - \xi^*$. The Newton iteration is

$$\xi_+ = \xi_c - F'(\xi_c)^{-1}F(\xi_c). \quad (4.2)$$

In (4.2) $F'(\xi_c)$ is the Jacobian matrix at the current iteration. Given sufficiently good data and sufficiently smooth nonlinearities the Newton iteration will converge quadratically, *i. e.*

$$\|e_+\| = O(\|e_c\|^2),$$

where $e = \xi - \xi^*$. In temporal integration applications the nonlinear iteration is terminated when

$$\|F(\xi)\| < \tau_a + \tau_r \|F(\xi_0)\|. \quad (4.3)$$

One does not compute the Newton step $s = -F'(\xi_c)^{-1}F(\xi_c)$ by using the inverse of the Jacobian but rather by solving (perhaps approximately) the linear equation

$$F'(\xi_c)s = -F(\xi_c). \quad (4.4)$$

For problems in three space dimensions the use of a direct method for solving (4.4) is impractical for reasons of both storage and computational cost. One must solve (4.4) by an iterative method. It is common [?, ?, ?] to terminate that linear iteration when the relative linear residual is small, *i. e.* when

$$\|F'(\xi_c)s + F(\xi_c)\| \cdot \eta_r \|F(\xi_c)\| \quad (4.5)$$

for some small η_r . (4.5) is called the inexact Newton condition. The parameter η_r is called the forcing term. In applications to temporal integration, one can use absolute residuals and gain some efficiency [?]. We can express convergence results for both approaches in terms of the termination condition

$$\|F'(\xi_c)s + F(\xi_c)\| \cdot \eta_a + \eta_r \|F(\xi_c)\| \quad (4.6)$$

If the initial iterate is near the solution, the nonlinearity is sufficiently smooth, and (4.6) holds for some η_a and $0 \cdot \eta_r < 1$ then the error in ξ_+ satisfies, [?, ?, ?],

$$\|e_+\| = O(\|e_c\|^2 + \eta_r \|e_c\| + \eta_a). \quad (4.7)$$

Clearly, if ξ_c is near ξ^* and η_r is sufficiently small the iteration will converge rapidly. However, solving the equation for the Newton step, (4.4), to very high accuracy may be wasteful, particularly if the initial iterate is far from the solution [?, ?].

4.1.2 The Linear Iteration and Preconditioning

If the linear equation (4.4) for the Newton step is solved by an iterative method, the overall iteration is called a Newton-Iterative method. The iteration (4.2) is called the *nonlinear iteration* or the *outer iteration*, and the iteration to solve the linear equation (4.4) is called the *linear iteration* or the *inner iteration*. In this work we use the Krylov method BiCGSTAB [?, ?, ?] as the linear solver with an additive Schwarz preconditioner. Such combinations were termed Newton-Krylov-Schwarz methods in [?] and that term is now common.

Preconditioning (from the right in our case), which is critical to good performance, replaces (4.4) with the equivalent system

$$F'(\xi_c)M\hat{s} = -F(\xi_c) \quad (4.8)$$

and then sets $s = M\hat{s}$. The preconditioning operator M is constructed so that (4.8) is easier to solve than (4.4).

4.1.3 Schwarz Preconditioners

The discretized elliptic problems that must be solved at each time step have the general form

$$\mathcal{R}(U_h, v) = 0, \text{ for all } v \in \mathcal{V}^h \quad (4.9)$$

where \mathcal{R} is the weak form of the nonlinear equation, \mathcal{V}^h is the space of test functions and $U_h \in \mathcal{V}^h$. \mathcal{R} is linear the second argument and nonlinear in the first. \mathcal{V}^h is the set of real-valued piecewise linear functions on the unstructured spatial mesh.

We begin by splitting the original physical domain - into subdomains - Ω_i and restricting the action of the differential operator to the subdomains. The division leads to the creation of new boundary pieces, called artificial boundaries, for the subdomains. Figure 1.1 depicts a one-dimensional domain - that is split into two subdomains, - Ω_1 and - Ω_2 . The artificial boundaries created by the split are Γ_1 and Γ_2 , with Γ_1 part of the boundary for - Ω_1 and Γ_2 part of the boundary for - Ω_2 .

Figure 4.1: Subdomain Splits in One-Dimension for Two Subdomains

When creating the subdomains from the original domain, one can require that the subdomains share an artificial boundary or one can allow the subdomains to overlap, as depicted in Figure 4.1. Subdomains overlap if part or all of the artificial boundary for one subdomain lies in the interior of an adjacent

subdomain. In Figure 4.1, the artificial boundary Γ_1 lies in the interior of the subdomain - 2, and similarly Γ_2 lies in - 1. In SWGW, the domains overlap with an overlap width of h , the width of a single element. Increased overlap is difficult to achieve when using unstructured grids. The domain decomposition methods used on overlapping domains are known as Schwarz methods; we concentrated on the additive Schwarz method. Discussions of Schwarz preconditioners may be found in [?].

The closure of the differential operator restricted to the subdomains requires that boundary conditions be placed on the artificial boundaries, as these boundaries were not part of the original problem formulation. Schwarz methods use zero Dirichlet boundary conditions on the artificial boundaries [?] because the algorithms incorporate error corrections on the artificial boundaries during the subdomain solves.

The one-level additive Schwarz preconditioner is a block Jacobi preconditioner. Let A be the discretization of the differential operator, and assume that the number of nodes of the discrete problem included in - i is n_i . Define the matrix R_i to be the (discrete) restriction operator for subdomain i , i.e., $R_i = \begin{bmatrix} 0 & I & 0 \end{bmatrix}$ where I is of size $n_i \times n_i$. If

$$B_i = R_i^T (R_i A R_i^T)^{-1} R_i$$

then the one-level additive Schwarz preconditioner is

$$M = \sum_{i=1}^p B_i \quad (4.10)$$

where p is the number of subdomains.

This preconditioner is implemented readily on a multiprocessor computer by assigning one or more subdomains to a processor. The amount of overlap between subdomains in SWGW is minimal; i.e., a single cell layer overlap is incorporated into the domain splitting. Because of this overlap and the zero Dirichlet conditions imposed on the artificial boundaries, there is no need for communication after the application of the one-level additive Schwarz preconditioner.

We found that the performance of the one-level method is inferior to two-level methods. We describe two such methods next. A two-level method requires a coarse mesh for either the entire computational domain or a coarse resolution of the entire domain.

Similarly to the one level preconditioner, we define a coarse mesh restriction operator R_0 and let $B_0 = R_0^T (R_0 A R_0^T)^{-1} R_0$. The two-level additive Schwarz preconditioner is

$$M = B_0 + \sum_{i=0}^p B_i \quad (4.11)$$

and the two-level hybrid II preconditioner [?, ?] is given by

$$M = B_0 + (I - B_0 A) \sum_{i=1}^p B_i. \quad (4.12)$$

In the work reported here we found that the alternative form

$$M = B_0 + \sum_{i=1}^p B_i (I - AB_0) \quad (4.13)$$

performed better.

To avoid generating and storing a separate coarse grid, we define coarse grid basis functions in terms of the already existing fine grid basis functions, an idea from algebraic multigrid [?]. One-dimensional examples of these coarse grid basis functions are depicted in Figure 4.2, with one coarse grid basis function defined per subdomain.

Figure 4.2: Coarse Grid Basis Function in One-Dimension

The coarse grid test and trial basis functions are formed by summing the fine mesh functions having support in a given subdomain. If $\{v_i\}$ is the nodal basis for \mathcal{V}^h and $\{D_I\}$ is the set of nodes in subdomain I , then the coarse mesh functions V_I are formed by

$$V_I = \sum_{i \in D_I} v_i.$$

The coarse grid problem can be constructed from the fine grid problem as follows. The fine grid problem is to find

$$\delta U_h = \sum_j \delta U_{h_j} v_j \quad (4.14)$$

such that

$$J(U_h, v_i) \delta U_h = -\mathcal{R}(U_h, v_i) \quad \text{for all } i, \quad (4.15)$$

where J is the Jacobian of \mathcal{R} . Then

$$B_0 = R_0^T [R_0 J(U_h, v_i) R_0^T]^{-1} R_0$$

where R is given by its action on a vector, i.e.,

$$(Ru)_I = \sum_{i \in D_I} u_i.$$

4.1.4 Parallel implementation

We assign one or more subdomains to each processor. Elements along the interprocessor boundaries are shared by those processors owning any of its nodes. The nodal information for the boundary elements must be communicated among processors so that each processor sees all of the shared elements.

The linear solver without preconditioning requires two types of communication: (1) an update of the nodal values along processor edges for each matrix vector product and (2) a global sum for each inner product. The additive Schwarz / block Jacobi preconditioning does not require any communication. The coarse mesh preconditioning also requires two types of communication. Initially, each processor forms part of the coarse matrix, and a global communication is used to assemble these parts on all of the processors. The coarse mesh problem is then solved redundantly on every processor. Each application of the preconditioner also requires the communication of the residual vector. Each processor sums its pieces of the residual vector, and a global communication is used to pass these pieces to all of the processors. The coarse preconditioner is then applied to the reduced residual vector. Each processor then expands the appropriate portion of the reduced vector and updates its portion of the full residual vector.

Chapter 5

Model Input

5.1 Boundary Condition Files

The boundary condition file contains many pieces of information necessary to perform simulations with ADH including specified pressure and velocity boundary conditions; Neumann and outflow boundary conditions; time step data; output controls; adaptivity controls; error tolerances; and maximum levels of mesh refinement. ADH accepts two types of boundary conditions: specified pressure or velocity (Dirichlet), and flux (Neumann). The code also has the capability of sending and receiving boundary information to/from outside programs.

Error tolerances are specified in the boundary condition file. Mesh refinement is governed by the specified error tolerance in the boundary condition file. In addition, the maximum levels of refinement must be specified by material type, giving the user additional control over the mesh adaptation.

A boundary condition file, *filename.bc*, for the ADH code contains a series of one line control cards. Cards are single line entries and cannot be wrapped across lines. The cards fall into eight basic categories: operational parameters, iteration parameters, material properties, boundary strings, solution controls/boundary conditions, time controls, output controls, and series. Operation Parameters control the operation of the code, the reserved memory space, type of problem being modeled, and the solver preconditioning arrangement. Iteration parameters control the iterative methods employed by the model. Material Properties define the flow and transport constants for each material in the model. Boundary conditions are set using string array cards defining the interior and surface boundaries of the problem, including node and face boundaries. Solution Controls specify the initial/boundary conditions. Time controls specify the time steps used to run the model. Output controls define the times at which the output is printed and Series cards are used to define various parameters. The different cards and their categories are shown in Table 5.1.

Each card consists of at least one character string identifying the type of card. It may then contain further character fields and/or numeric data fields. There

Operation Parameters (Sec. 5.1.1)	OP NS	Navier-Stokes
	OP INC	Incremental Memory
	OP TRN	Transport Quantities
	OP BLK	Blocks per processor
	OP PRE	Preconditioner
	OP THT	θ time weighting
Iteration Parameters (Sec. 5.1.2)	IP NIT	Non-Linear Iterations
	IP NTL	Non-Linear Tolerance
	IP MIT	Maximum Linear Iterations
	IP FNI	Forced Non-Linear Iterations
	IP FLI	Forced Linear Iterations
Materials (Sec. 5.1.3)	MP EV	Eddy Viscosity
	MU	Viscosity
	MP G	Gravitational Acceleration
	MP RHO	Density
	MP NCE	Non-conservative calculations
	MP TMN	Momentum Stabilization Coefficient
	MP TCN	Continuity Stabilization Coefficient
	MP UO	Reference Velocity
	MP LO	Reference Length
	MP RO	Reference Density

Boundary Strings (Sec. 5.1.4)	NDS	Node String
	FCS	Face String
	BNS	Border Node String
	GNS	Ghost Node String
Solution Controls (Sec. 5.1.5)	DB VEL	Dirichlet - Velocity
	DB PRS	Dirichlet - Pressure
	NB VEL	Neumann - Flow
	NB PRS	Neuman - Pressure
	OB VEL	Outflow Boundary
	DB EXT	Dirichlet - Pressure (externally supplied)
	OB EXT	Outflow Boundary (externally supplied)
Time Control (Sec. 5.1.6)	TC TO	Start Time
	TC IDT	Time Series
	TC TF	Final Time
	TC ADP	Adaptive Time Control
	TC NDP	Non-Adaptive Time Control
Series	XYS	X-Y Series Cards
	XYT	Curve Fit Tolerance
Output Controls (Sec. 5.1.7)	OC	Output Control Series
	END	Signifies End of BC file

Table 521: Cards

are two important points to note about this file. First, the leading 6 columns are reserved for character field keywords ONLY. All numeric data MUST start in column 7 or later. As an example, consider the line of input

```
MP_K12.62.62.60.00.00.0
```

This input would result in two character fields being read. One would have the value “MP” and the other, the value “K” as intended. The following incorrect line

```
MP_K12.62.62.60.00.00.0
```

would result in fields containing the values “MP” and “K 1”. It is important to note that the parser cannot handle lines more than 150 characters wide. An input file template is provided in Table A.9.

5.1.1 Operation Parameters

Each Operation Parameter card consists of two character fields and may contain one numeric field. Operational parameter cards are identified by an “OP” in the first field. OP cards control the type of system is being modeled. An OP NS card is used to specify Navier-Stokes flow modeling.

The code allocates memory as needed during the run to store the additional elements and nodes created during the refinement process. The memory is allocated in blocks. The size of the block is specified by the user on the OP INC card. If the specified number is too small, the program will continually seek additional memory, slowing the run time of the program. Alternately, if the number is too large, the program will require excess memory not needed to run the code.

The preconditioner for the linear solver and the manner in which it is implemented are specified by the OP PRE and the OP BLK cards. The first card specifies the preconditioner used. The integer can be 0,1,2,or 3 for various preconditioning schemes. The second card defines how many blocks per processor are to be used in the preprocessing. These are subdividing the problem to perform a direct solve on each block and the total group of all blocks can be used to perform a coarse grid solve. Which of these options is used is specified by the OP PRE choice. In this case, the 2 indicates two-level Additive Schwartz preconditioning using 10 blocks per processor.

```
OP NS
OP INC 2000
OP PRE 2
OP BLK 10
```

After finding a flow solution, an associated transport problem can be solved.

The number of transported quantities (*#transport-equations*) is given on an OP TRN card. The OP TRN card is a required input card. If the problem does not involve transport, zero (0) quantities are specified on the OP TRN card. In addition, if transport equations are not being modeled, no transport properties or boundary conditions may be specified. An error message will be displayed if transport properties are included in the input file but no transport quantities have been specified. The format of the OP cards is found in Table A.1. The following card specifies one transported quantity. Presently, no transported items are supported in the Navier-Stokes solver.

OP TRN 0

5.1.2 Iteration Parameters

There are three iteration parameter cards that must be specified by the user. Iteration parameter cards are identified by an “IP” in the first field. An IP NIT card specifies the maximum number of non-linear iterations. An IP NTL card specifies the convergence tolerance for the non-linear iterations. An IP MIT card specifies the maximum number of linear iterations for each non-linear iteration. At the maximum number of iterations specified on the IP NIT or IP MIT cards, if the convergence is not sufficient the calculations will fail. Another option is available for each of these cards. These function like these two cards but if the maximum iteration count is reached the calculations are accepted and ADH proceeds. The IP FNI card then is for the non-linear iteration maximum and IP FLI is for the linear iteration maximum. The format of these cards is specified in Table A.2.

IP NIT 5
IP NTL 1.0E-3
IP MIT 20000

5.1.3 Materials

Material Property cards are identified by the designation “MP”. There will be a set of cards for each material type in the model. Material properties are separated into two groups: one for the coefficients in the flow equations and the other for coefficients in the transport equations. Each group contains a set of refinement control cards. Refinement may be adjusted to independently follow the error in the flowfield and transport equations. For all of these cards, the first two fields contain character strings, specifying the type of card (MP) and the specific parameter (K, SS, FVS, SAT, KR, TVS, DF, DPL, DPT, RCT, POR, TOR, ML, FRT, or TRT as defined below). The third field is an integer field containing the material number to which the values apply, (*mat#*). The formats for these cards are listed in Tables A.3 and A.4.

Flow Parameters

Five cards are needed to specify flow parameters: hydraulic conductivity (K), material storage coefficient (SS), volumetric flow source (FVS), pressure-saturation (SAT), and relative conductivity (KR). The SAT and KR cards define the relationship among suction pressure, saturation, and relative conductivity. The values on these cards are only important for problems involving partially saturated flow.

Hydraulic Conductivity, K The hydraulic conductivity or system stiffness is an important input parameter for groundwater flows. Hydraulic conductivity has units of L/T and is a function of the medium and the fluid flowing through it. The hydraulic conductivity, K, comes from Darcy's law in the form of

$$v = -K \frac{dh}{dl} \quad (5.1)$$

Where:

$$\begin{aligned} h &= \text{Hydraulic head} \\ \frac{dh}{dl} &= \text{Hydraulic Gradient} \end{aligned}$$

The hydraulic conductivity of sands and gravels is typically higher than for silts and clays. The hydraulic conductivity is expressed as a tensor in the following form:

$$\begin{array}{ccc} K_{xx} & K_{xy} & K_{xz} \\ K_{xy} & K_{yy} & K_{yz} \\ K_{xz} & K_{yz} & K_{zz} \end{array}$$

The six values in the upper right quadrant of the tensor are entered on the MP K card in the following order: K_{xx} , K_{yy} , K_{zz} , K_{xy} , K_{xz} , K_{yz} . If the hydraulic conductivity is independent of the direction of measurement, the formation is termed *isotropic*. In the isotropic case, $K_{xx} = K_{yy} = K_{zz}$ and $K_{xy} = K_{xz} = K_{yz} = 0$. The MP K card is required. The units chosen for the hydraulic conductivity must be consistent with all other units in the program. This becomes particularly important for the units of time as they must correspond with the time units specified by the user in the time control portion of the run.

Material Storage Coefficient, SS The material storage coefficient, or storativity, is defined as the volume of water released by a unit volume of soil due to a unit decrease in hydraulic head. The volume of water released is a function of two factors: 1) the volume of water displaced by the compaction of the soil caused by the increase in effective stress and 2) the expansion of the water due to the decrease in pressure. The storativity has the units of 1/L, defined as the volume per unit volume per unit decrease in head. The MP SS card must be included. The storativity is specified by a single numeric input on this card.

Volumetric Flow Source, FVS For model problems containing volumetric sinks or sources, a flow source card, MP FVS, is used to specify the strength of a source or sink. A single numeric value should be specified on the MP FVS card in units of L^3/T . Again, it is important that all units are consistent throughout the input of the material properties (the user should verify that the units used for the volumetric flow source are consistent with those used for the hydraulic conductivity). The following cards represent material 1 with an isotropic hydraulic conductivity of 2.6×10^{-3} , a storativity coefficient of 0.1 and no volumetric flow source/sink.

MP K	1	2.6e-3	2.6e-3	2.6e-3	0.0	0.0	0.0
MP SS	1	0.1					
MP FVS	1	0.0					

Suction-Saturation Relationship, SAT It is necessary to define the relationship between suction pressure and saturation. This relationship is required for both saturated and partially saturated flow; however, when the flow is completely saturated, the value of saturation will be a constant set equal to one (1.0), representing fully saturated media. For partially saturated flow, it is preferable to use saturation data from laboratory testing; however, if this information is not available, the suction-saturation curve can be generated using a modified version of the van Genuchten formula (van Genuchten, 1980) as shown in equation 5.2.

$$\bar{S} = (1 + |\alpha h|^\beta)^{-m} \quad (5.2)$$

where

$$\begin{aligned} \bar{S} &= \text{Relative Saturation} \\ h &= \text{Suction or Negative Pore Water Pressure Head} \\ \beta &= \text{van Genuchten Parameter} \\ \alpha &= \text{van Genuchten Parameter} \end{aligned}$$

and

$$m = (1 - \frac{1}{\beta}) \quad (5.3)$$

The van Genuchten parameters are not fundamental parameters, instead they are parameters used to curve fit experimental data. The van Genuchten formula is in terms of the suction head, h , with the units of L . The product of αh is dimensionless, with α having the dimensions of $1/L$.

The relative saturation, S , is defined by the following equation:

$$\bar{S} = \frac{S - S_r}{S_s - S_r} \quad (5.4)$$

where

$$\begin{aligned}
S &= \text{Current Value of the Degree of Saturation} \\
S_r &= \text{Residual Saturation} \\
S_s &= \text{Degree of Saturation if Soil is Fully Saturated} \\
(\text{Note: } S_s &\text{ is typically } 1.0)
\end{aligned}$$

Typical values for the van Genuchten parameters are listed in Table 5.2.

<i>Soil Type</i>	α [1/cm]	β	S_r
Sand	0.145	2.68	0.17
Loamy Sand	0.124	2.28	
Sandy Loam	0.075	1.89	
Loam	0.036	1.56	
Silt	0.0016	1.37	0.20
Silty Loam	0.020	1.41	
Sandy Clay Loam	0.059	1.48	
Clay Loam	0.019	1.31	
Slity Clay Loam	0.010	1.23	
Sandy Clay	0.027	1.23	
Silty Clay	0.005	1.09	
Clay	0.008	1.09	0.84

Table 5.2: van Genutchen Parameters

Figure 5.1 shows the suction-saturation relationship for three typical materials (sand, silt, and clay). ADH will develop a spline-fit curve based on the data entered by the user. In ADH, the suction-saturation relationship is represented by a X-Y series referenced on the MP SAT card. The series should be entered with values of the suction pressure in the first field and values of the saturation in the second field.

When modeling problems that do not involve partially saturated flow, such as confined aquifer regions, the relationship between suction pressure and saturation should be assigned a constant value of 1.0.

Relative Conductivity, KR For saturated flow, the hydraulic conductivity is constant and is not a function of the pressure head. However, the KR card is required and the value of the series should be set to 1.0.

In partially saturated flow, the hydraulic conductivity is a function of the saturation which may be related to the pressure head. ADH requires the user to define the relationship between the relative conductivity and the suction pressure, or negative pore water pressure, for each material. Like the suction-saturation relationship, the relative conductivity will be entered as a series referenced by the MP KR card for each material type. ADH will then spline-fit the

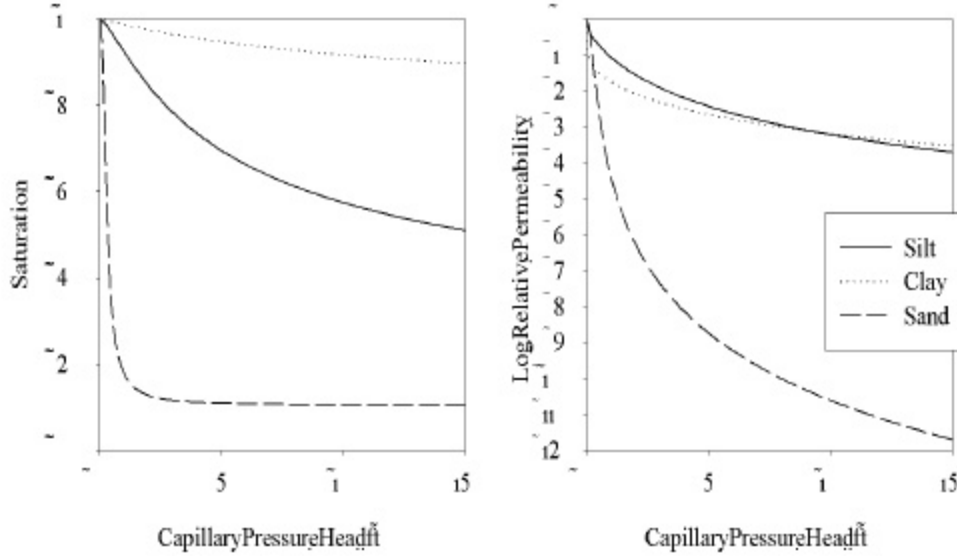


Figure 5.1: Suction-Saturation Relationship for Typical Sand, Silt, and Clay

data to develop the curve representing the relative conductivity for each material. Again, it is preferable to obtain data on the relative conductivity for each soil type from laboratory testing. However, if this information is not available, the relationship can be established using the van Genuchten formula.

van Genuchten defines the relationship for the relative conductivity in the following equation:

$$K_r = \bar{S}^{(0.5)} [1 - (1 - \bar{S}^{(\frac{1}{\gamma})})^\gamma]^2 \quad (5.5)$$

where

$$\begin{aligned} K_r &= \text{Relative Conductivity} \\ \gamma &= \frac{1}{\beta} \end{aligned}$$

Using the van Genuchten parameters, as listed in Table 5.2, the relationship between the suction pressure and the relative conductivity can be established. Figure 5.2 shows the relationship between suction pressure and relative conductivity for the three soils listed in Table 5.2.

Transport Properties

To specify transport equations, a set of seven cards is required for each transported quantity in each material type. These cards include mass source/sink

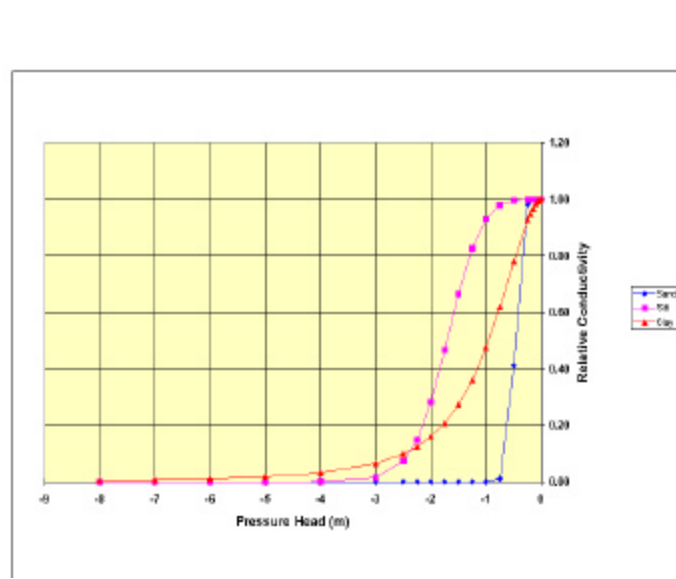


Figure 5.2: Relative Conductivity for Sand, Silt, and Clay

strength, molecular diffusion, dispersivity, reaction rate, porosity, tortuosity, and mesh refinement. The first three fields of these cards follow the form stated above for hydraulic conductivity storativity, and volumetric sinks. The fourth field is an integer string identifying the identification number of the transport constituent, (*tran#*).

Mass Source/Sink Strength, TVS For problems that include a mass sink or source of a transport quantity, the strength of the source/sink is specified on an MP TVS card. The strength would be specified by the user in the units of M/T .

Molecular Diffusion, DF Diffusion in solution is the process whereby ionic or molecular constituents move under the influence of their kinetic activity in the direction of their concentration gradient (Freeze and Cherry, 1979). A diffusion coefficient for each transport material must be entered on the MP DF card. The units for the diffusion coefficient is L^2/T . Diffusion coefficients for ions typically found in groundwater (Na^+ , K^+ , Mg^{2+} , Ca^{2+} , Cl^- , HCO_3^- , SO_4^{2-}) range from 1×10^{-9} to $2 \times 10^{-9} m^2/sec$. (Freeze and Cherry, 1979). The third field of the MP DF card references the number of the transport quantity. The fourth field is a real number representing the molecular diffusion coefficient.

Dispersivity, Longitudinal (DPL) and Transverse (DPT) Longitudinal and transverse dispersivity relate to the mixing and dilution of a solute in

a groundwater flow system. Dispersion is the cumulative result of mechanical mixing during fluid movement and the thermal kinetic energy of the solute particles. Mechanical mixing results from variations in the local velocity. ADH uses values of the longitudinal and transverse dispersivity to account for mechanical dispersivity. The dispersion of the solute in the direction of the main flow lines is termed longitudinal dispersivity. The dispersion in the direction normal to the flow is the transverse dispersivity. The units of dispersivity are Length. Dispersivity parameters are specified on a pair of cards. Longitudinal dispersivity, D_L , is specified on an MP DPL card and transverse dispersivity, D_T , on an MP DPT card.

Reaction Rate When modeling problems with transported quantities, an MP RCT card is required to input the reaction rate of the transport quantity. The third field of the card references the material type identification number. The fourth field represents the transport quantity and the fifth field references the reaction rate.

Porosity, POR The porosity ϕ of each material must be specified on a MP POR card. Porosity for a unit volume of soil is defined by the following equation:

$$\phi = \frac{V_V}{V_T} \quad (5.6)$$

Where

$$\begin{aligned} V_V &= \text{Volume of the Voids} \\ V_T &= \text{Total Volume} \end{aligned}$$

Porosity is calculated as a volumetric ratio, making it a unitless quantity. Ideally, the porosity of the soil used in the model should be determined by performing laboratory soils testing on field samples. However, if lab data are not available, it is important to choose a reasonable value for the porosity of the soil type chosen. Davis (1969) compiled typical ranges of porosity for a wide range of soils.

Tortuosity, TOR Tortuosity relates to the path through which a fluid flows through a soil matrix, specifically the orientation, shape, and length of capillary tubes. Tortuosity is determined by the shape and arrangement of the soil particles and takes into account the shape of the pores through which the fluid flows. The tortuosity factor is defined as the ratio of the length of the curvilinear microscopic flow path to the length of the linear macroscopic flow path (Mesri, 1980). The tortuosity, τ , is a dimensionless quantity (L/L) and is specified on a MP TOR card.

Retardation Parameter, RD The retardation parameter, or retardation factor, is entered on the MP RD card. The retardation parameter is specific to

the material and transport quantity. In the third field of the *MP RD* card, the material number is entered. The transport quantity is entered in the fourth field, and the retardation parameter is entered in the fifth field.

MP	TVS	1	1	0.0
MP	DF	1	1	0.315
MP	DPL	1	1	10.0
MP	DPT	1	1	1.0
MP	RCT	1	1	0.0
MP	POR	1	1	0.4
MP	TOR	1	1	0.7
MP	RD	1	1	1.0

Mesh Refinement

An *MP ML* card is used to specify the maximum levels of mesh refinement, or the total number of times that an original element may be split within a material type. Refinement can be turned off in a material by specifying zero (0) as the maximum levels of refinement. When refinement is on, the solution error tolerance is given on an “RT” card. There is one card for the flow problem (*MP FRT*) and one for each of the transport systems (*MP TRT*). In each case, the “RT” card specifies the refine error tolerance. If the solution error on an element exceeds the refine error tolerance given on an “RT” card, the element is split.

MP	ML	1	5
MP	FRT	1	1.0e10
MP	TRT	1	1.0e0

Different material types can have different levels of refinement. Some experimentation with the error tolerance is usually necessary to gain the desired level of refinement. These tolerances and the maximum number of refinement levels may also be changed as the model advances.

5.1.4 Boundary Strings

For most groundwater problems, boundary data includes Dirichlet data on or interior to the domain, line sinks and sources through the domain, and flux data (Neumann) through a region of the domain surface. Each of these boundary conditions is applied to a “string” of element nodes or faces. Each component of a string is input on a card that specifies the string type and node or element face it contains. There are three types of boundary strings: node, face, and edge. Complete strings are input on multiple cards with one node, face or edge per card. Cards may be input in any order and cards for different node strings may be interspersed. See Table A.5 for details on the format of these cards.

Node Strings

Dirichlet data are specified on node strings. These can be made up of boundary and/or interior nodes as the problem requires. The identifier for this card is NDS. On each card, the node string number (*string#*) is followed by a node number.

Face Strings

Neumann data are specified across face strings. The identifier for this card is FCS. The card lists the identifier, string number, element number, and then an element face number (*face#*). Currently, mixed or Robin boundary conditions are not supported.

NDS 1037 2
FCS 1008 1 1

5.1.5 Solution Controls

Solution development is controlled through the specification of the initial and boundary conditions and the time step parameters. Dirichlet boundary conditions are specified on a DB card, and Neumann data on a NB card. These cards contain either a FLW or TRN specifier in the second field to signify if they apply to the flow or transport equations respectively. The third field of the DB, and NB cards specify the node string to which they apply. For transport quantities, the fourth field of the card specifies the ID number of the transport constituent to which they apply. The last field of the DB and NB cards references a time-boundary condition series number. This XY series allows the user to change the boundary condition during the simulation. Values for the Dirichlet or Neumann boundary conditions are linearly interpolated by ADH; therefore, it is important to define the series with a linear interpolation in mind. Any boundary without an explicit boundary condition receives a no-flux boundary condition by default.

For outflow boundaries, a OB card is used. The second field of an OB card references flow (FLW) or transport (TRN). The third field references a boundary string, representing the outflow face, and the fourth field contains the ID number of the transport quantity. See Table A.7 for a complete description of these cards.

DB FLW 1 1 4
NB FLW 2 1 5
DB TRN 1 2 6
NB TRN 2 2 7
OB TRN 7 1

5.1.6 Time Controls

Evolution of the solution is determined by a group of five cards with the Time Control specifier TC. The start time is specified on a TC TO card. The TC IDT references a time series in the third field that will control the time steps. The final time, at which the run will terminate, is specified on a TC TF card. The final time does not have to correspond with the largest value in the time series.

For adaptive time control, a TC ADP card is used. Adaptive time control allows the model to refine the time steps from those specified by the user in the time series. The model may choose finer time steps than those specified but will not adapt the time steps to a larger time step value.

A TC NDP card is used for non-adaptive time steps. In this case, the code will only refine the time steps for stability purposes, not for accuracy. Time steps will only be reduced if the model fails to converge for the current time step.

TC TO 0.0
TC IDT 5
TC TF
TC ADP
TC NDP

5.1.7 Output Control

An OC card causes the solution to be printed at startup and at each specified time step. The OC card references a series that controls the output data. These are output as data set files.

An END statement is used at the end of the boundary conditions file. The code will read the boundary conditions file through to the END statement. Any information in the boundary conditions file after the END statement will not be read as input to the run. Reference Table A.8 for a full description of the above cards.

OC 3
END

5.2 Hot Start File

The hot start file, *filename.hot* is used to specify two types of model data: initial conditions and scale factors. Initial conditions consist of piezometric heads and concentrations of transported materials. Scale factors in the hot start files are applied to the hydraulic conductivity and dispersion tensors.

Field data is often available and used as a starting point for many problems. The field data can be specified as the initial conditions used in the flow and

transport equations for a specific problem. These data are specified in the “hot-start” file. The GMS provides a simple interface for entering field data. This data is entered into a scatter point data file and interpolated to the problem mesh.

A hot start file is a required part of a model; however, it may be a zero-length file (empty). For simple problems a cold start simulation (initial conditions of 0.0) may be sufficient. Cold start problems in ADH are handled by leaving the hot-start file blank. If data types are not specified default values will be supplied. A set of predefined dataset names is used to declare data types, shown below.

IH	Initial Head
IC	Initial Concentration
K	Hydraulic Conductivity
D	Molecular Diffusion

For the IC and D datasets, a number is appended to the NAME card to specify which of the transported materials it applies to. The parser will accept any amount of whitespace (including none) between the name and number.

The hydraulic conductivity and dispersivity tensors are used to specify anisotropic properties as cell-based data in each model material. This is adequate for problems where each material has a constant conductivity or dispersion. Tensor scale factors are node-based data and are useful to model layers where the properties have a gradient within the layer, but the same degree of anisotropic behavior. They are also useful for blending the material properties together at material boundaries. This is especially useful as strong jumps in material properties can make the model mathematically stiff and difficult to solve.

The datasets used for the hot-start file can be generated with the GMS. A standard GMS 3-D mesh data set format is used in the hot start file. Multiple data sets are exported from the GMS and copied to the hot start file in any order. If a dataset is not supplied for one or more of the parameters, ADH will assign default values to all the cells for that parameter. The tensor scalars K and D are given a default value of 1.0. Initial conditions on the head and any solute concentrations assume a value of 0.0.

Chapter 6

Using the GMS with ADH

The GMS is a powerful tool for mesh generation and visualization of results. In addition, the GMS can be used for establishing boundary conditions and initial conditions. The GMS builds 3-D tetrahedral meshes by building a 2-D mesh and projecting that mesh through a domain. A horizontal 2-D mesh is constructed using triangular elements, quadrilateral elements, or a mixture thereof. These elements are then projected through the depth of the domain to generate 3-D prismatic or hexahedral elements respectively. Tetrahedral elements are then developed with a refining process. Since ADH performs automatic mesh refinement, initial meshes should typically be much coarser than those used in other modeling systems. When using the GMS, tetrahedral elements should be produced using the “fine” refinement setting to avoid directional grid bias. “Fine” refinement will place a new node midway between existing neighbor nodes, in effect splitting every edge. For this reason, a starting grid should be twice as coarse (fewer elements) as the desired initial tetrahedral mesh.

6.1 Problem Solving with ADH

ADH is a stand alone application; however, it is designed to be used in conjunction with the GMS for problem solving. A future GMS release will have a full interface to ADH, similar to the GMS FEMWATER interface. Until the release of the interface, it is necessary to use the available GMS tools in conjunction with some manual steps to develop models. Two simple models are presented in the following sections with most of the model construction steps explained in detail. This is intended to demonstrate how to use ADH in conjunction with the GMS. For a full understanding of the capabilities of the GMS, it is recommended that the user perform the full suite of tutorials in the GMS.

6.1.1 Traveling Plume

One way to test the accuracy of the ADH algorithm is to develop a problem that models the migration of a contaminant plume in a larger domain of soil. If a contamination plume is defined in a soil domain, a flowfield can be developed by defining a time-dependent head drop. The introduction of the head drop will cause the contaminant plume to move in the direction of the flow field.

The following example illustrates the traveling plume. The large soil domain can be easily represented by a rectangular cube of soil. A smaller cube of chemical solute can be centered in the domain at the top surface of the soil cube, as shown Fig. 6.1. The following section details the mesh generation process for this sample problem using the GMS.

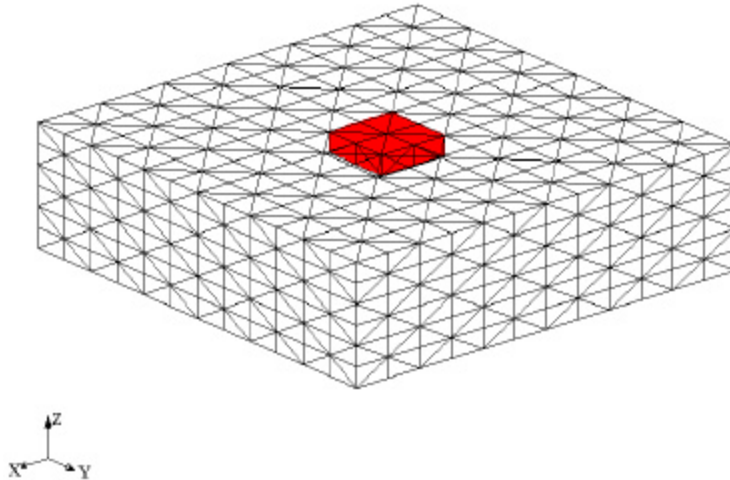


Figure 6.1: Initial shape and location of solute.

Mesh Generation in the GMS

In order to construct a three dimensional mesh, it is necessary to construct a two-dimensional mesh and project it through a given field.

Construction of the 2-D Mesh Construction of this model begins in the GMS by entering the “2D Mesh” mode. Using the “Create Nodes” tool, place four arbitrary points in the mesh. Using the “Select Nodes” tool, select one of the points and edit the coordinates of that point in the “Edit” box to $x = -60$; $y = -60$. (Note: it may be necessary to select the z-coordinate to ensure the y-value is entered correctly. Be sure to verify the location of each point in

the edit box). Continue this procedure with the three remaining nodes, editing the coordinates of each point to $(\pm 60.0, \pm 60.0)$. Select the “Frame Image” tool to show all four points on the main screen (This can also be done by selecting “Frame Image” under the view menu).

The four points on the main screen represent the outer boundary of the soil domain through which the contaminant plume will travel. Since we will not use mesh refinement in this example, it is necessary to add additional nodes so that the mesh is not too coarse. To add the additional nodes to the system, select any two consecutive nodes. Under the “Modify Mesh” menu, select “Node Interp Opt...” and specify six intervals in the node string. This options selects the number of nodes that will be generated between the two selected nodes. Repeat this process on the remaining sides of the domain, subdividing each side with additional nodes. Now that the boundary of the domain is defined, the domain is ready to be filled in with elements. Select all the nodes by dragging a box around the elements or enter the “Edit” menu and use the “Select All” function (Be sure you are using the “Select Nodes” Tool).

Once all of the nodes are selected, enter the “Build Mesh” menu and select the “Rectangular Patch” option. A text box will prompt the user to choose between triangular and quadrilateral elements. Choose the “Quadrilateral Elements” option. The text box in the “Edit” window will prompt the user to select two corners of the domain in the counter clockwise direction. Follow the directions in the text box. At the completion of this step, a 2-D mesh will appear on the screen, as seen in Figure 6.2.

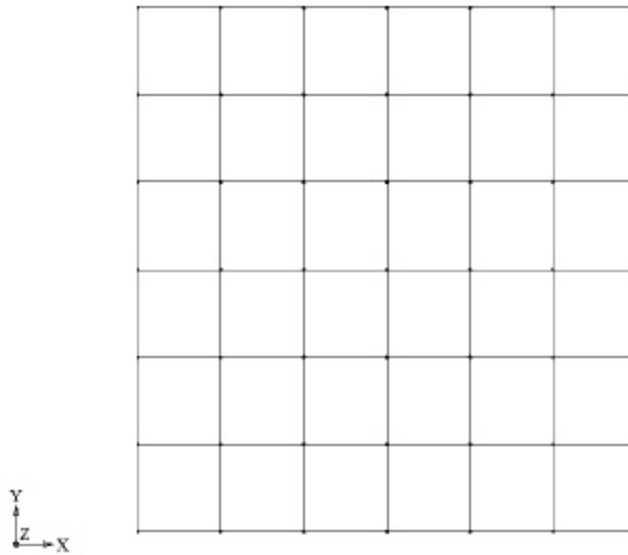


Figure 6.2: Completed 2-D Mesh

Construction of the Boreholes This 2-D mesh is now projected through a domain to produce the 3-D mesh. This is accomplished by defining a set of boreholes to the desired “depth” of the domain. We will define four boreholes at the outer corners of the domain (This is an arbitrary choice. When defining actual field problems, borehole location will be selected upon site specific data). Switch to the “Borehole Module” and enter the “Boreholes” menu. From the “Boreholes” menu, select the “Borehole Editor.” Once in the borehole editor, select “New Hole.” The hole can be given a name for reference. This may be useful for later editing of the borehole file. Once the new hole is named, the x, y, and z coordinates of the top and bottom of the hole must be defined. For the four holes, the x and y coordinates should correspond to the corners of the 2-D mesh, with coordinates of $(\pm 60.0, \pm 60.0)$. For each borehole, enter a value of zero (0) for the top of the hole and a value of (-30) at the bottom of the borehole. The GMS will automatically assign default material numbers (a value of zero will be defined at the top surface and a value of 1 will be defined at the bottom surface). Change the material number at the top surface to a value other than zero. Be sure that the material color on the right side of the borehole editor changes by selecting the material number box for the bottom of the borehole. Once the four holes are defined, select “OK.”

This set of boreholes can be saved as a *filename.bor* file. It is recommended that you save this file at this point so that the boreholes can be used if changes to the mesh are required at a later point. (This will save regeneration of the boreholes).

Generation of the 3-D Mesh The boreholes will now be selected to generate the 3-D mesh. Change the view from 2D mesh mode to 3D mesh mode. Using the “Select Segments” tool, select any one of the boreholes. Go to the “Boreholes” menu and select the “Regions->3D Mesh...” option. The user will be prompted to select the number of layers of elements in the mesh. Your choice of layers will be used in the determination of the initial number of elements in the 3-D Mesh. For this example, specify two layers of elements. At the completion of this step, a 3-D mesh will appear on the screen.

Refining the Mesh It is now necessary to refine the elements of the mesh. Enter the 3-D Mesh module and select “Refine Elements” from the “Meshes” menu. The GMS will prompt that this function will delete all boundary condition. Select “OK” at the prompt. In the dialog box that appears, click on “Refine all 3-D mesh elements”, “All elements -> tets”, and “Fine refinement”.

When the GMS refines the mesh, it may not number elements and nodes consecutively, skipping some values in its numbering sequence. Although this is not a problem with mesh generation, this will cause an error in ADH. Therefore, it is good practice to always renumber the mesh. In the 3-D Mesh module, select the “Select Boundary Faces” tool. While holding down the *ctrl* key, select an element on one of the boundary faces. The selected face will appear highlighted. Enter the “Meshes” menu and select “Renumber.”

To check that the mesh is correctly renumbered, select “Get Info...” from the “File” menu. If the number of nodes and maximum node ID have the same value as well as the number of elements and maximum element ID, the grid is valid. If not, it must be renumbered to remove the “blanks” in the node/element numbering sequence.

Saving the Final Mesh The mesh is now complete and should be saved. In the “File” menu, select the “Save” option. Enter a filename and select the “Update” option. This will assign the root filename for all files associated with this mesh (including the 2-D mesh, and borehole file). Once saved, it may be preferable to remove data that is not included in the 3-D mesh. For this example, enter the 2-D Mesh module. Enter the “Edit” menu and choose “Select All.” Once all the 2-D mesh information is selected, press the *delete* key or select “delete” from the “Edit” menu. Follow this procedure for the borehole data by entering the Borehole module, selecting all of the data, and deleting the data.

Generating the Boundary Conditions File

After generating the mesh, it is necessary to build the boundary conditions file. This file will contain the operation controls, iteration parameters, boundary strings, solution controls, time controls, and output controls.

Selecting the Boundary Condition Faces Boundary conditions are created assigning boundary conditions to a node/face string. It is convenient to use the GMS to select these strings. The GMS will group nodes/faces by assigning them a string number. This information can be used to assign the boundary conditions to the node string in ADH. The process of selecting the boundary face in the GMS is merely a method of specifying and grouping nodes/faces with the same boundary condition.

Since the example problem models flow and transport through the mesh, boundary conditions will be specified on four sides of the cube-shaped mesh. The top and bottom of the cube-shaped mesh will represent no-flow boundary conditions.

After preparing the mesh, enter the 3-D mesh mode of GMS and select the Boundary Node tool. Select the “Oblique View” option from the view menu. Select one face of the box by selecting a node on the desired face while holding down the *ctrl* key. Select the “Assign Node/Face BC ...” from the FEMWATER menu. (Note: A completed ADH menu will be available in the next version of GMS. The use of the FEMWATER menu is strictly for the beta version of ADH.) A dialog box will appear where a head or flux boundary condition can be specified. Select the “Head” option to specify a Dirichlet boundary condition and specify a constant value for the head. The value entered in the GMS is arbitrary because the actual value of the head will be specified in the ADH boundary conditions file. However, it is important to assign a *different* number to each boundary condition string and note to which face the value of head corresponds. For this example, the first face will be

selected and set to a constant head value (1.0), representing face 1 of the cube, as seen in Fig. 6.3.

Next, select the boundary face adjacent to the originally selected face and repeat the process of assigning an arbitrary head constant to the boundary face. (Be sure that the arbitrary constant is a different than the value used for the first face and that the value has significance to the user for identifying the face at a later time). For this example, a constant head of 2.0 will be used (to represent face 2).

Rotate the oblique view so that the opposite side of the mesh is facing the user. This can be accomplished by entering the “View” menu and selecting the *View Angle* option. Enter a bearing angle of 220 degrees.

Select the boundary face opposite the first face selected and assign another arbitrary constant head (making sure the value is different from the first two chosen values). Repeat this process for the fourth side of the mesh.

Once the boundary condition faces have been specified, save the file by selecting “Save Simulation” in the FEMWATER module. This is a different option than the “Save” item under the “File” menu of GMS. It is critically important that you do not save over the .3dm file that was saved in the “File” menu. This can be avoided by using a different name for the boundary conditions file or de-selecting the 3dm option on the FEMWATER save menu.

Operation Controls Once the boundary faces are selected, the boundary conditions file can be generated. The *filename.bc* file will be created in a text editor. The operation controls should be located at the top of the file. For this example, an *OP GW* card will be used to specify that the problem models groundwater flows. This card will be followed by an *OP INC* card to specify the incremental memory allocation. For this example, a value of 2000 will be sufficient. The next card will be an *OP TRN* card specifying the number of transport quantities. For this example, a value of 1 should be entered in the third field as only one transport quantity is moved through the mesh. The operation controls will have the following format:

OP GW
OP INC 2000
OP TRN 1

Iteration Parameters Three iteration parameters will be specified by the user, the maximum number of non-linear iterations, *NIT*, the tolerance of the non-linear iterations, *NTL*, and the maximum number of iterations, *MIT*. For this example problem, the maximum number of non-linear iterations will be set to 20, the non-linear tolerance will be set to 1.0×10^{-6} , and the maximum number of linear iterations will be set at 200. The iteration parameters will have the following format:

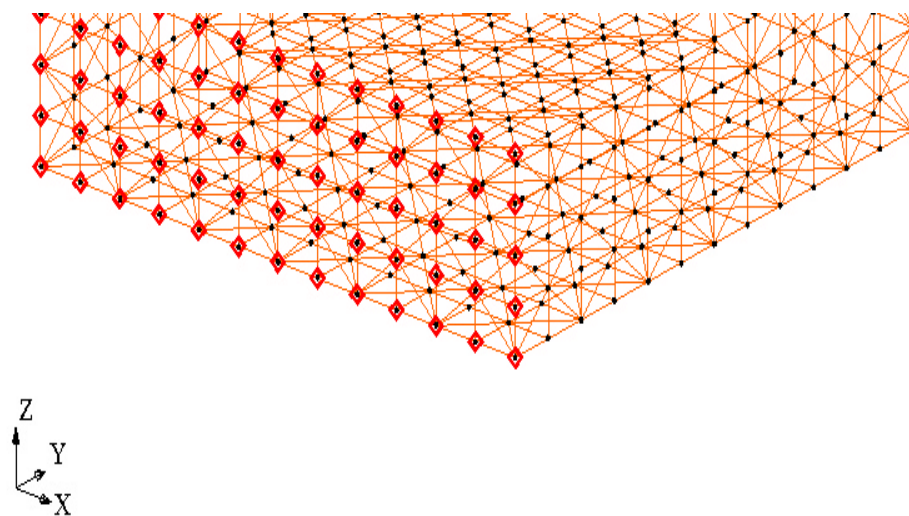


Figure 6.3: Selection of the Boundary Nodes for Face 1

IP	NTL	1.0E-6
IP	NIT	20
IP	MIT	200

The linear convergence tolerance is automatically set to single precision machine zero. This typically has a value that is on the order of 1.0×10^{-7} which implies that the non-linear tolerance can not be set any lower than about 1.0×10^{-6} .

Material Properties The next section of the boundary conditions file will contain information about the material properties. The hydraulic conductivity will be entered on an *MP K* card. For this example a hydraulic conductivity of 1.0×10^{-3} cm/sec (0.11808 ft/hr) , typical for a sand material, will be used. The material storage coefficient will be entered on the *MP SS* card. For this example the Material Storage Coefficient will be set to zero. Although the example does not contain a volumetric source/sink, an *MP FVS* card is included. For this example, the strength of the volumetric flow source will be set to zero.

Two cards are required to describe the behavior of the flow and transport in the partially saturated zones, the *MP SAT* and the *MP KR* cards. The pressure saturation curve will be referenced by the *MP SAT* card. The fourth field of the card references a series that will be input below, in this case, series 3. The relative conductivity of the material will be referenced by the *MP KR* card. Like saturation, this card references a series representing the relationship between the suction pressure and the material's relative conductivity. In this case, series 4 is specified.

These material properties will be entered in the following format:

MP K	1	0.11808	0.11808	0.11808	0.0	0.0	0.0
MP SS	1	0.0					
MP FVS	1	0.0					
MP SAT	1	3					
MP KR	1	4					

Eight cards are required to describe the material transport parameters. An *MP DF* card is used to specify the molecular diffusion coefficient. The third field of the card references the material identification number, the fourth field references the transport quantity number, and the fifth field represent the molecular diffusion coefficient. For this example, we will assume that there is no molecular diffusion and enter a value of zero.

The longitudinal and transverse dispersivity must be entered for problems involving transported materials. Both a *MP DPL* and *MP DPT* card are included. For simplicity, these values will be set to zero for this example.

The reaction coefficient must be entered on a *MP RCT* card. For this example, it is assumed that the transport material is non-reactive. The material porosity will be entered on the *MP POR* card. For this example a value of 0.5 will be used for the porosity of the sand. The tortuosity for the material will be entered on the *MP TOR* card. For this example, the tortuosity will be set equal to 1.0.

The retardation parameter will be set on the MP RD card. For this example a value of 1.0 will be used. A MP TVS card will be used to specify the transport volumetric source. For simplicity it will be assumed that there is no transport volumetric source in this example problem.

```
MP DF  1 1 0.0
MP DPL 1 1 0.0
MP DPT 1 1 0.0
MP RCT 1 1 0.0
MP POR 1 1 0.5
MP TOR 1 1 1.0
MP RD  1 1 1.0
MP TVS 1 1 0.0
```

Three other *MP* cards are required to specify the level of mesh refinement. An *MP ML* card is used to specify the maximum levels of mesh refinement. For this simple example, the mesh refinement will not be used and the maximum levels of refinement will be set to zero. Although the mesh refinement is not activated, it is still necessary to include an *MP FRT* or a *MP TRT* card; however, their specified value is inconsequential. It may be advisable to specify reasonable values for the tolerances so little modification must be made to the boundary conditions file if the user decides to use the mesh refinement option on subsequent runs of the code. The remaining material property cards will have the following format:

```
MP ML 1 0
MP FRT 1 0.28
MP TRT 1 1 1.0e0
```

Boundary Strings The next section of the boundary conditions file generates the NDS cards that are used to group node strings. The boundary faces selected in the GMS can now be imported into the ADH boundary conditions file. During the generation of the boundary conditions in the GMS, a FEMWATER boundary conditions file was created with the extension *.3bc*. This file contains the information that allows the user to assign the appropriate boundary conditions in the ADH file. For this example problem, the beginning of the file contains the following lines:

```

XY1 1 1 0 0 0 0.000000000000000e+00 constant
0.000 4.000000000e+00
XY1 2 1 0 0 0 0.000000000000000e+00 constant
0.000 3.000000000e+00
XY1 3 1 0 0 0 0.000000000000000e+00 constant
0.000 1.000000000e+00
XY1 4 1 0 0 0 0.000000000000000e+00 constant
0.000 2.000000000e+00

```

The node string series number (which begins with the series indicator XY1) is assigned a constant head on the line following the series. In this case, series 1 is assigned a constant head of 4. When in the GMS, a constant head of 4 represented boundary face 4. Therefore, node string 1 represents face 4 of the cube. Likewise, node string 2 represents face 3, node string 3 represents face 1 and node string 4 represents face 2.

All of the node strings are also contained in the *.3bc* file in the following format:

DB1	845	1
DB1	844	2
DB1	843	1
DB1	841	1
DB1	839	1
DB1	821	3
DB1	820	1
DB1	819	2
DB1	795	3
DB1	794	3(etc...)

In the FEMWATER *.3bc* file, the first field specifies a Dirichlet Boundary Condition (as opposed to a Neumann or flux boundary condition), the second field corresponds to the node number, and the third field references a boundary string.

ADH uses a very similar format for assigning the boundary strings; however the input in the first field must be changed to read *NDS*. Copy the boundary strings into the ADH boundary conditions file and make the appropriate changes to the first field (changing DB1 to NDS). The resulting strings will have the following form:

NDS	845	1
NDS	844	2
NDS	843	1
NDS	841	1
NDS	839	1
NDS	821	3
NDS	820	1
NDS	819	2
NDS	795	3
NDS	794	3(etc...)

Solution Controls Solution controls are the mechanism by which the geometric boundary of the mesh is assigned the value of the boundary condition. The solution controls are used to specify Dirichlet boundary conditions and Neumann boundary conditions for both flow and transport problems. Any boundary without explicit boundary conditions receives a no-flux boundary condition.

For this example, a Dirichlet boundary condition for the flow problem will be set on all four faces of the cube-shaped mesh. The boundary condition will remain constant; however, the boundary condition, representing total head, will be lowered on two sides of the cube to create a flow field. The solution control cards will assign an X-Y series to each of the four boundary strings representing the boundary faces. The first field in the card specifies whether the boundary condition represents a Dirichlet or a Neumann boundary condition. The second field specifies flow or transport. The third field specifies the string number (as designated in the previous step) and the fourth field specifies a series that sets the value for the boundary condition.

For the first face, with a boundary string number of 3, the boundary condition will be set with XY-series number 1. For the second face, with a boundary string number of 4, the boundary condition will be set with XY-series number 2. For face number 3, with a boundary string number of 2, the boundary condition will be set with XY-series number 7. For face number 4, with a boundary string number of 1, the boundary condition will be set with XY-series number 8.

The following cards will be entered in the boundary conditions file:

DB FLW 3 1
DB FLW 4 2
DB FLW 2 7
DB FLW 1 8

For transport of the plume, only boundary faces 3 and 4 will require boundary conditions. These faces will set a zero concentration on the inflow face. The third field of the card specifies the boundary string, the fourth field will represent the transport quantity (in this case transport quantity 1) and the fifth field will be assigned a transport boundary condition represented by XY-

series 9. Face 4, also corresponding to transport quantity 1, will be assigned a transport boundary condition represented by series 10. The following cards will be used in the ADH boundary conditions file:

DB	TRN	2	1	9
DB	TRN	1	1	10

Time Controls The evolution of the solution will be determined by a group of cards with the time control specifier *TC*. These cards will specify the initial time, a time series that will be used throughout the run to control the time step, the final time, and adaptive time control. For this example, non-adaptive time control will be used. The initial time will be set to zero. The final time will be set to 200 and a time step series will be referenced by XY-series 5. The time controls in the boundary conditions file will be in the following format:

TC	TO	0.0
TC	IDT	5
TC	TF	200.0
TC	NDP	

Output Control The output control *OC* card references a series when the output of the run will be written to a file. For this example, the output series will be number 6 and the output control card in the boundary conditions file will have the following format:

OC	6
----	---

Series Referenced by the Boundary Condition File Cards Once the cards in the boundary conditions file are complete, it is necessary to enter the series that are referenced in the cards. All series have the following format:

XY1	3	30	0	0	0
-----	---	----	---	---	---

The first field, *XY1* signifies a series. The second field references the series number, in this case, series 3. The third field references the number of points in the series, in this case 30. The remaining three fields were developed for use in the GMS. They have no significance to the output of ADH; however, they should remain zeros in the boundary conditions file.

Series 1 sets the flow boundary conditions for the face 1 of the cube. The x-value represents time and the y-value represents a value of total head for the Dirichlet boundary condition. It is important to note that ADH will perform

a linear curve fit from the data points entered in this file. The following series will set the total head equal to -2 for face 1 of the cube.

XY1	1	3	0	0	0
0.0			-2.0		
100			-2.0		
200.0			-2.0		

Series 2 sets the boundary condition for the second face. Like the boundary conditions for the first face, we want to set a Dirichlet boundary condition of -0.75 . The following series will result:

XY1	2	4	0	0	0
0.0			-0.75		
100			-0.75		
200.0			-0.75		

Series 3 defines the relationship between the suction pressure and the saturation of the material. For this example, the suction-saturation curve has been generated using the van Genuchten formula for a typical sand. The x-value in the series represents the suction pressure and the y-value represents the saturation. The format of the series is as follows:

XY1	3	20	0	0	0
-8.2			0.170291492		
-6.396			0.171142836		
-5.904			0.171774481		
-5.412			0.172862287		
-4.92			0.174830425		
-4.428			0.178606515		
-3.936			0.186378015		
-3.444			0.203755204		
-2.952			0.246283709		
-2.46			0.356774641		
-1.968			0.607383809		
-1.476			0.894394331		
-1.312			0.946973259		
-1.148			0.976803528		
-0.984			0.991316013		
-0.82			0.997323946		
-0.656			0.999370855		
-0.164			0.999999923		
-0.0328		1			
0		1			

Series 4 defines the relationship between the suction pressure and the relative conductivity. For this example, the van Genuchten formula was used to determine the relative conductivity for a typical sand. The values in the x-series represent the suction pressure and the values in the y-series represent the relative conductivity. The file will be in the following format:

XY1	4	20	0	0	0
-8.2		9.18E-11			
-6.396		4.59E-09			
-5.904		1.62E-08			
-5.412		6.36E-08			
-4.92		2.85E-07			
-4.428		1.49E-06			
-3.936		9.41E-06			
-3.444		7.48E-05			
-2.952		0.000776927			
-2.46		0.010283908			
-1.968		0.124850087			
-1.476		0.599075309			
-1.312		0.763295568			
-1.148		0.879162016			
-0.984		0.947024262			
-0.82		0.980546273			
-0.656		0.994356019			
-0.492		0.998857354			
-0.328		0.999879187			
0		1			

Series 5 is the time series that will dictate the time steps for the run. The x-value in the series represents the time. The y-value in the series represents the time steps. The time series will have the following format:

XY1	5	9	0	0	0
0.000		0.5			
2.000		1.000			
10.000		2.000			
20.000		5.000			
60.000		10.000			
100.000		20.000			
200.000		50.000			
600.000		100.000			
1200.000		200.000			

Series 6 sets the output times when data will be written to a file. The x-value represents the time when the user wants the output to be written to a file. The y-value is a dummy variable and does not have significance. It is suggested that the user leave this value set to zero.

The file will have the following format:

XY1	6	20	0	0	0
0.0	0.0				
1.0	0.0				
2.0	0.0				
3.0	0.0				
4.0	0.0				
10.0	0.0				
20.0	0.0				
30.0	0.0				
40.0	0.0				
50.0	0.0				
60.0	0.0				
70.0	0.0				
80.0	0.0				
90.0	0.0				
100.0	0.0				
120.0	0.0				
140.0	0.0				
160.0	0.0				
180.0	0.0				
200.0	0.0				

Series 7 sets the Dirichlet boundary conditions for the third face of the cube. The x-value represents time and the y-value represents a value of total head for the Dirichlet boundary condition. A total constant head of zero 0 will be used to represent a constant flow source at the inflow face of the cube.

XY1	7	3	0	0	0
0.0	0.0				
100	0.0				
200.0	0.0				

Series 8 sets the boundary condition for the fourth face. Like the boundary conditions for the third face, a Dirichlet boundary condition of 0.0 will be set for a constant flow source at the inflow face. The following series will result:

XY1	8	3	0	0	0
0.0			0.0		
100			0.0		
200.0			0.0		

Ending the Boundary Conditions File An END statement must be located at the end of the boundary conditions file. Information can be contained in the file after the end statement; however, it will not be included in the run.

END

Generating the Hot Start File

The hot start file will contain information on the initial concentrations, the initial heads, and the weighting factors for the hydraulic conductivity and the dispersion tensors.

Generating the Initial Concentration File, IC The plume is specified as an initial concentration occupying a cube in the middle of the mesh. Specifying the plume concentration is done in the GMS using by entering data in the as scatter points and interpolating the points onto the mesh. However, this may not be an easy task if the node concentrations are entered manually. The nodes around the contamination “box” must be identified and concentrations must be assigned to these nodes. For this example, a contamination concentration of 0.17 will be used. Additional nodes must be specified with a zero concentration. The GMS will interpolate the concentrations on the nodes surrounding the specified nodes. Because of the interpolation schemes used by the GMS and the volume of the mesh, it will be necessary to set many points throughout the mesh to a zero concentration so that the interpolated data shows that the concentration of the contaminant is confined to the desired small area within the mesh.

Scatter Point Data File A scatter point data file will be used to enter the concentrations at specific nodes in the mesh. Scatter point data files are text files that can be generated in any editor or spreadsheet in the following format:

ID	X	Y	Z	Concentration
----	---	---	---	---------------

The first field contains an identification number (*ID*) that is inconsequential; however it is convenient to assign an identification number equal to the node number so errors can be easily tracked. The x, y, and z values correspond to the coordinate system in the mesh. The concentration at each node is specified in the fifth field in units consistent to all other units in the model.

For this example, it is preferable that the mesh have an initial concentration of 0.0 at every point except for the region occupied by the plume, which will

have a concentration of 0.17. It is possible to generate the scatter point data file by identifying the nodes in the GMS where concentrations will be specified and generating the file in a text editor with these nodes and concentrations. However, to model the situation described above, it will require the user to specify many points with a zero concentration (in excess of 200) so that the plume does not “spread” tremendously due to the interpolation scheme.

Alternatively, it is possible to export a scatter point data file from the GMS which represents all points in the mesh. This file can be imported into any editor or spreadsheet to assign the concentration values at the desired nodes. From the 3-D mesh mode, with the mesh displayed on the main screen in the GMS, enter the *Mesher* menu and choose the “Mesh->Scatter Points” option. The GMS will prompt for a filename and ask if the user wants to delete the existing 3-D mesh data. Any file name is appropriate for the scatter points. Do not delete the 3-D mesh data. It is now necessary to save the scatter point data file so that modifications to the file can be made in a text editor or a spreadsheet. Under the *File* menu, choose the “Save” option. Un-select all default options except the *3D Scatter* option and name the scatter point data file. The file will have a *.xyz* extension.

Open the *filename.xyz* file in a spreadsheet or text editor. The file will contain four main columns. The first column contains the node number corresponding to the nodes in the mesh. The second, third, and fourth column represent the x-, y-, and z-coordinates of the node. The first five lines of the file will contain the following information:

```
SCAT3D
BEGSET
NAME {filename}
ID 10797
IXYZ 845
```

These lines can be deleted from the file and replaced with headings above each main column. The resulting file should have the following format:

```
ID X Y Z
1 6.000000000000000e+01 -6.000000000000000e+01 0.000000000000000e+00
2 6.000000000000000e+01 -5.000000000000000e+01 0.000000000000000e+00
3 6.000000000000000e+01 -4.000000000000000e+01 0.000000000000000e+00
4 6.000000000000000e+01 -3.000000000000000e+01 0.000000000000000e+00
5 6.000000000000000e+01 -2.000000000000000e+01 0.000000000000000e+00
```

A fifth column must be added to the file to assign the concentrations. Depending on the capabilities of the text editor the user prefers, it may be advantageous to perform this function in a spreadsheet. The fifth column should be titled “Concentration” and a value of zero (0.0) should be entered on each line (for each node in the mesh).

The resulting file will have the following format:

ID	X	Y	Z	Concentration
1	6.0000000000e+01	-6.0000000000e+01	0.0000000000e+00	0.0
2	6.0000000000e+01	-5.0000000000e+01	0.0000000000e+00	0.0
3	6.0000000000e+01	-4.0000000000e+01	0.0000000000e+00	0.0
4	6.0000000000e+01	-3.0000000000e+01	0.0000000000e+00	0.0
5	6.0000000000e+01	-2.0000000000e+01	0.0000000000e+00	0.0

This scatter point data file now represents a concentration of zero at every nodal point in the mesh. Save this file as a text file with an *.xyz* extension.

To define the zone of the mesh with the contaminant plume, start by changing to the top view of the mesh. Using the “Select Elements” tool, drag-select all of the elements not located in the eight cells surrounding the center node, and hide the surrounding elements with the “Hide Elements” tool. This will be accomplished in several steps of selecting elements and hiding them.

Once all of the elements are hidden except for the center eight elements, Switch to a side view and hide all elements below the top-most layer. The remaining cube that is visible on the screen will contain the initial plume. Select the oblique view so all the nodes will be visible. Go to “Display Opts...” under the **Display** menu. Turn on “Node numbers” and “Elements” with the “All elements” option. The node numbers of the elements comprising the initial plume will now be displayed on the screen, as shown in Figure 6.4

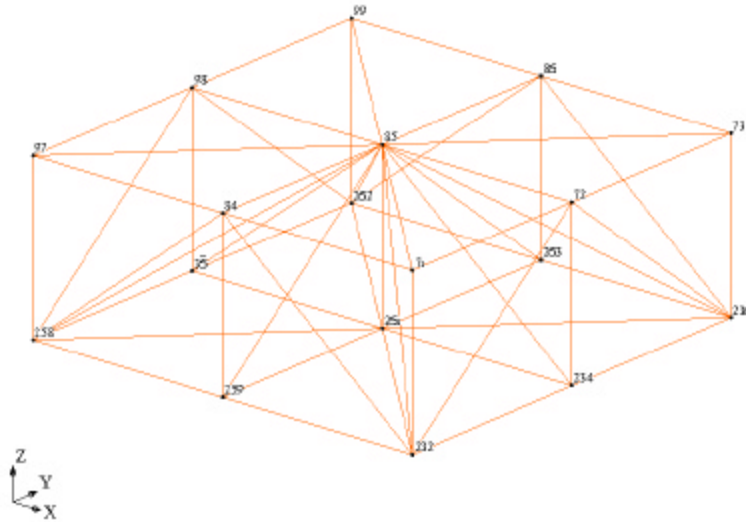


Figure 6.4: Selection of elements of solute plume.

Record all 18 node identification numbers (note: the mesh may have to be rotated to see all numbers clearly).

Go back to the scatter point data file and locate the nodes on the contaminant plume (the numbers in the ID column will correspond to the node numbers identified in the previous step). Change the value of the concentration at each node from 0.0 to 0.17. Once this step is complete, resave the scatter point data file. The entries in the scatter point data file representing the contaminant plume will have the following format:

ID	X	Y	z	Concentration
97	-10	-10	0	0.17
98	-10	0	0	0.17
99	-10	10	0	0.17
84	0	-10	0	0.17
85	0	0	0	0.17
86	0	10	0	0.17
71	10	-10	0	0.17
72	10	0	0	0.17
73	10	10	0	0.17
258	-10	-10	-7.5	0.17
260	-10	0	-7.5	0.17
262	-10	10	-7.5	0.17
259	0	-10	-7.5	0.17
261	0	0	-7.5	0.17
263	0	10	-7.5	0.17
232	10	-10	-7.5	0.17
234	10	0	-7.5	0.17
236	10	10	-7.5	0.17

Once the scatter point data file is complete, it must be imported into the GMS for interpolation to the mesh. Switch to the 3-D Scatter Point Module. In the *File* menu select “Import.” Select “Multiple Data Sets 3-D” from the screen menu. The GMS will then ask for the file name. Choose the scatter point data file, *filename.xyz*. The scatter points will appear highlighted on the mesh.

To interpolate the scatter point data to the mesh, select “Interpolation Options” under the *Interpolation* menu. Choose the “Inverse Distance Weighted” and select the “Options” button. In the “Nodal Function” window choose the “Constant (Shephard’s Method)” option. Note: for other problems, it may be advantageous to choose alternative interpolation schemes. Select “OK” and turn on the “Truncate Values” option. Select “OK.” Now go back into the “Interpolation” menu and select the *...to 3D mesh* option. The GMS will prompt for the data set name. Enter “IC1” (Corresponding to the initial concentration for transported material number 1) and select “OK.”

The initial concentration has now been interpolated to the mesh. To view the concentration on the mesh, go back into the 3D mesh mode and select the *Display Options* tool. Select the “Fringes” option and “OK” in the two menus

to get back to the GMS screen. Now use the shading tool to shade the object. The resulting contaminant plume should appear as shown in Figure 6.5.

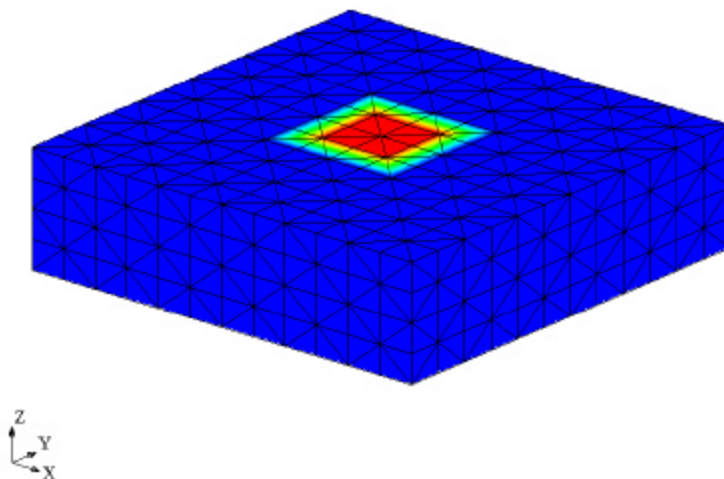


Figure 6.5: Initial Plume

Exporting the Initial Concentration File The initial concentration file must be exported from the GMS for incorporation into the hot start file. From the *Data* menu, select “Data Browser.” Choose the “Export” option and select “Save.” The GMS will prompt for a filename. Name the file *IC1*, representing the initial concentration of transport quantity number 1. The output file will have an *.dat* extension.

Generating the Initial Head File, IH To specify the initial heads, data can be interpolated to the mesh in a very similar way to the initial concentrations. However, if no initial head file is specified, ADH will assign the default value of a constant total head of zero at every node in the mesh. For this example, the default value will be used and the IH file will remain empty.

Combining the files The hot start file contains all the files for the initial concentration, the initial head, and the weighting factors for the hydraulic conductivity and the molecular diffusion. Typically these files would be copied into one master file (in any order) and saved with the title *filename.hot* extension. For this example, we will assume that the hydraulic conductivity and molecular diffusion do not have weighting factors and will be given the default value by

leaving the file blank. Since the initial head, hydraulic conductivity, and molecular diffusion files are empty for this example, the user must re-name the file *IC1.dat* to *filename.hot*. Prior to saving the final version of the hot start file, open the file in the editor and insert a space between *IC* and *1* in the NAME line of the file (line number 6). Save the final version of the hot start file.

6.1.2 Running pre_ADH

Once all of the input files needed for ADH are generated (the *.3dm*, *.bc*, and *.hot* files) the user should run *pre_ADH* to create the input files for ADH. The 3-D mesh, boundary conditions, and hot start file should all have the same root filename. To run *pre_ADH*, simply type the command “*pre_ADH filename*”. The code will output messages to the user showing that each file is properly read into the code. If there are any errors in the input files, *pre_ADH* will display an error message.

6.1.3 Running ADH

Once *pre_ADH* has been successfully executed, the user can run ADH using the command *ADH filename*.

In Fig. 6.6, results are presented for the example. The first figure, Fig. 6.6a shows the initial plume with a reference “cross-hair”. The plume has spread out evenly through numerical dissipation in Fig. 6.6b. This is due to the stabilization routine in ADH. The center of the plume is still relatively intact and is seen to have moved diagonally from the center cross-hair. The final figure, Fig. 6.6c shows the plume to once again be positioned at the center of the domain although a significant of spreading has occurred.

6.2 References

- Howington, S. E., Peters, J. F., Illangasekare, T. H., and R. S. Maier (1997) “Discrete Network Modeling for Field Scale Flow and Transport Through Porous Media,” Technical Report 97-21, Waterways Experiment Station, Vicksburg, MS.
- Jenkins, E. W., Schmidt, J. H., Stagg, A. R., Howington, S. E., Berger, R. C., Hallberg, J. P., Kelley, C. T., and M. D. Tocci, “Newton-Krylov-Schwarz Methods for Hydrology,” in review.
- Kelley, C. T. (1995) *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia.
- Liu, A. and B. Joe (1995) “Quality Local Refinement of Tetrahedral Meshes Based on Bisection.” *SIAM Journal of Scientific Computing*, 16(6), pp. 1269-1291.
- Schmidt, J. H. (1995) *A 3-D Adaptive Finite Element Method for Transport Processes*, Master’s Thesis, The University of Texas at Austin, Austin, TX.

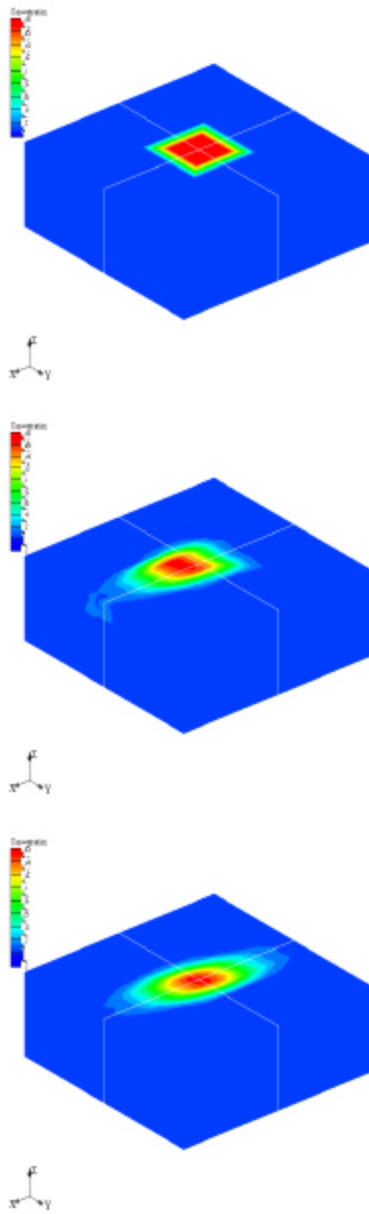


Figure 6.6: Movement of a solute plume without dispersion or diffusion.

Singh, V. P. (1996) *Kinematic Wave Modeling in Water Resources, Surface Water Hydrology*. John Wiley and Sons, New York.

Tompson, A. F. B. and L. W. Gelhar. (1990) "Numerical Simulation of Solute Transport in Three-Dimensional, Randomly Heterogeneous Porous Media." *Water Resources Research*, 26(10), pp. 2541-2562.

Tompson, A. F. B., Ashby, S. F., Falgout, R. D., Smith, S. G., Fogwell, T. W., and G. A. Loosmore. (1994) "Use of High Performance Computing to Examine the Effectiveness of Aquifer Remediation." In *Proceedings of the X International Conference on Computational Methods in Water Resources*, Vol. 2, Dordrecht, Kluwer Academic.

Appendix A

Constructing Model Files

This appendix gives an overview of the three files needed to run ADH. Details and explanations of each of the input parameters is given in the body of the report and in the sample problems. Three files are required for running ADH: the 3-D Mesh File, the Boundary Conditions File, and the Hot Start File. The generation of each of these files and the components of the files are described in the following sections.

A.1 3-D Mesh Files

The three dimensional mesh files needed for ADH are generated completely within the GMS. Once the mesh has been generated in the GMS, the file will be used in ADH without modification. The filename given to the mesh file, having an extension *.3dm*, will serve as the root name for all ADH input files. Details on mesh generation can be found in the example problems contained within this text or in the GMS reference manual.

A.2 Boundary Conditions

The boundary conditions file contains a series of cards that represent the operation controls, iteration parameters, material properties, boundary strings, solution controls, time controls, and output controls. The following tables contains all of the possible boundary condition file cards and a description of their input.

Navier-Stokes Problems			
Field	Type	Value	Description
1	char	OP	Card type.
2	char	NS	Specifies Navier-Stokes Problem

Incremental Memory			
Field	Type	Value	Description
1	char	OP	Card type.
2	char	INC	Parameter.
3	int	$\# > 0$	Incremental Memory Allocation

Time Step Weighting			
Field	Type	Value	Description
1	char	OP	Card type.
2	char	THT	Parameter.
3	int	$1 \geq \# \geq 0$	θ .

Block Specification for Pre-conditioner			
Field	Type	Value	Description
1	char	OP	Card type.
2	char	BLK	Parameter.
3	int	$\# > 0$	Number of blocks per processor, used to perform pre-conditioning

Pre-conditioner Selection			
Field	Type	Value	Description
1	char	OP	Card type.
2	char	PRE	Parameter.
3	int	$3 \geq \# > 0$	Prec value 0 No pre-conditioning 1 one level Additive Schwartz pre-conditioning 2 two level Additive Schwartz pre-conditioning 3 two level Hybrid pre-conditioning

Table A.1: Operation Parameter Cards

Table A.3: Material Property Cards

Eddy Viscosity			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	EV	Parameter.
3	int	≥ 1	Material type ID number.
4	real	$\# > 0$	E_{xx}
5	real	$\# > 0$	E_{yy}
6	real	$\# > 0$	E_{zz}
7	real	$\# > 0$	E_{xy}
8	real	$\# > 0$	E_{xz}
9	real	$\# > 0$	E_{yz}

Non-conservative Calculation Option			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	NCE	Parameter. If present calculations are non-conservative, otherwise, the calculations are conservative.

Viscosity			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	MU	Parameter.
3	real	$\# \geq 0$	Uniform background viscosity.

Gravitational Acceleration			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	G	Parameter.
3	real	$\# \geq 0$	Value of gravity induced acceleration.

Density			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	RHO	Parameter.
3	real	$\# \geq 0$	density.

Table A.3: Material Property Cards (cont.)

Momentum Stabilization Coefficient			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	TMN	Parameter.
3	real	$0 \geq \# \geq 0.5$	Galerkin Least Squares stabilization coefficient associated with the momentum equations.

Continuity Stabilization Coefficient			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	TCN	Parameter.
3	real	$0 \geq \# \geq 0.5$	Galerkin Least Squares stabilization coefficient associated with the continuity equation.

Reference Velocity			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	U0	Parameter.
3	real	$\# \geq 0$	Reference Velocity.

Reference Length			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	L0	Parameter.
3	real	$\# \geq 0$	Reference length.

Reference Density			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	R0	Parameter.
3	real	$\# \geq 0$	Reference Density.

Non-linear Iterations (option 1)			
Field	Type	Value	Description
1	char	IP	Card type.
2	char	NIT	Parameter
3	int	≥ 1	Number of non-linear iterations per time step. If at NIT the tolerance is not satisfied, ADH will reduce the time step and recalculate.

Non-linear Iterations (option 2)			
Field	Type	Value	Description
1	char	IP	Card type.
2	char	FNI	Parameter.
3	int	≥ 1	Number of non-linear iterations per time step. If at FNI the tolerance is not satisfied, ADH will accept the solution and proceed to the next time step.

Non-linear Tolerance			
Field	Type	Value	Description
1	char	IP	Card type.
2	char	NTL	Parameter.
3	int	≥ 0	Tolerance for Non-Linear Equations

Linear Iterations (option 1)			
Field	Type	Value	Description
1	char	IP	Card type.
2	char	MIT	Parameter.
3	int	≥ 1	Maximum number of linear iterations per non- linear iteration by the iterative solver. If the internal linear toler- ance (0.01 * NTL) is not met at MIT the solution stops.

Linear Iterations (option 2)			
Field	Type	Value	Description
1	char	IP	Card type.
2	char	FLI	Parameter.
3	int	≥ 1	Maximum number of linear iterations per non-linear iteration by the iterative solver. If the internal tolerance (0.01 * NTL) is not met at FLI the solution will proceed to the next nonlinear it- eration.

Table A.2: Iteration Parameter Cards

Refinement Levels			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	ML	Parameter.
3	int	≥ 1	Material type ID number.
4	int	$\# \geq 0$	Maximum number of refinement levels.

Flow Refinement Tolerances			
Field	Type	Value	Description
1	char	MP	Card type.
2	char	FRT	Parameter.
3	int	≥ 1	Material type ID number.
4	real	$\# \geq 0$	Error tolerance for the refinement terms.

Table A.4: Material Meshing Control Cards

A.3 Hot Start File

The **Data Calculator** is useful for generating simple hot-start files. For most problems, the hydraulic conductivity and molecular diffusion weight factors do not need to be specified. The default value of 1.0 is usually valid. If not, it is fairly simple to generate constant datasets for these and/or the initial conditions.

Open the 3-D mesh model in the GMS. In the **Data Calculator**, select the **elevation** file for that mesh as “Operand 1”. This is a default data set associated with every mesh. Set the “Operator” to subtraction and declare “Operand 2” to also be the **elevation** file. Specify “Result” to be some set such as **base** and click the “Compute” button.

Next load **base** in as “Operand 1”. Use the addition operator and declare “Operator 2” with the scalar value of an initial condition. Specify the appropriate key in “Result”, i.e. **IH**, **IC 1**, etc. and compute.

Once this procedure is complete for all the necessary data sets, go into the **Data Browser** and delete **base**. Use the “Export” function with the “All data sets” option to write the data out to *filename.hot*.

This method is valid for generating data sets with a constant value. It is also a starting point for data sets with mostly constant values. The few variances can be corrected by hand. For more complicated systems, such as those matching detailed field data, the file will have to be constructed using the GMS geostatistical tools or the GMS interpolation from scatter point data tools.

Node Strings			
Field	Type	Value	Description
1	char	NDS	Card type.
2	int	≥ 1	String ID number.
3	int	≥ 1	ID number of a node with a Dirichlet condition.

Face Strings			
Field	Type	Value	Description
1	char	FCS	Card type.
2	int	≥ 1	String ID number.
3	int	≥ 1	Element number.
4	int	≥ 1	ID number of a face with a Neumann condition.

Border Node String			
Field	Type	Value	Description
1	char	BNS	Card type.
2	int	$\# \geq 1$	Node number list, in order, of nodes which have data that are to be supplied to an external program.

Ghost Node String			
Field	Type	Value	Description
1	char	GNS	Card type.
2	int	$\# \geq 1$	Node number list, in order, of nodes which have data supplied by an external program.

Table A.5: String Structures

Dirichlet Velocity Boundary Condition			
Field	Type	Value	Description
1	char	DB	Card type.
2	char	VEL	Parameter.
3	int	≥ 1	String ID number
4	int	$\# \geq 1$	Series ID number for x-velocity component.
5	int	$\# \geq 1$	Series ID number for y-velocity component.
6	int	$\# \geq 1$	Series ID number for z-velocity component.

Dirichlet Pressure Boundary Condition			
Field	Type	Value	Description
1	char	DB	Card type.
2	char	PRS	Parameter.
3	int	$\# \geq 1$	String ID number.
4	int	$\# \geq 1$	Series ID number.

Dirichlet External Pressure Boundary Condition			
Field	Type	Value	Description
1	char	DB	Card type.
2	char	EXT	Parameter.
3	int	$\# \geq 1$	String ID number, pressure defined by external program.

Neumann Pressure Boundary Condition			
Field	Type	Value	Description
1	char	NB	Card type.
2	char	PRS	Parameter.
3	int	$\# \geq 1$	String ID number.
4	int	$\# \geq 1$	Series ID number.

Neumann Flow Boundary Condition			
Field	Type	Value	Description
1	char	NB	Card type.
2	char	VEL	Parameter.
3	int	$\# \geq 1$	String ID number.
4	int	$\# \geq 1$	Series ID number containing the flow (in velocity) out of this boundary.
5	real	$\# \geq 0$	The Darcy-Weisbach friction boundary parameter for this boundary.

Table A.6: Initial and Boundary Conditions

Outflow Boundary Condition			
Field	Type	Value	Description
1	char	OB	Card type.
2	char	VEL	Parameter.
3	int	$\# \geq 1$	String ID number identifying the boundary face for which flow is allowed to pass out of the domain.

Outflow External Boundary Condition			
Field	Type	Value	Description
1	char	OB	Card type.
2	char	EXT	Parameter.
3	int	$\# \geq 1$	String ID number identifying the boundary face for which flow information is supplied by an external program. If this flow is into the model this flow is enforced, if the externally supplied flow is out then the face is treated as an outflow boundary.

Table A.7: Initial and Boundary Conditions (continued)

Starting Time			
Field	Type	Value	Description
1	char	TC	Card type.
2	char	T0	Parameter.
3	real	#	Starting time of the model.
Number of Time Steps			
Field	Type	Value	Description
1	char	TC	Card type.
2	char	NTS	Parameter.
3	int	# > 0	Number of time steps to be run.
Time Step Size			
Field	Type	Value	Description
1	char	TC	Card type.
2	char	DT	Parameter.
3	real	# > 0	Length of timestep (Δt).
Running the Model			
Field	Type	Value	Description
1	char	RUN	Solve the model according to the current settings.
Output			
Field	Type	Value	Description
1	char	OC	Output the solutions and grids for each time step.
Stopping the Model			
Field	Type	Value	Description
1	char	END	Close the model.

Table A.8: Solution Control Cards

```

OP MN #nodes
OP ME #elements
OP SAT
OP TRN #transport_equations
IP MIT max_iterations_per_time_step
MP K mat# Kxx Kyy Kzz Kxy Kxz Kyz
MP SS mat# S
MP FVS mat# source_strength
MP ML mat# max_levels_refinement
MP FRT mat# error_tolerance_for_refinement
MP FUT mat# error_tolerance_for_unrefinement
MP TVS mat# tran# source_strength
MP DF mat# tran# Dxx Dyy Dzz Dxy Dxz Dyz
MP DPL mat# tran# DL
MP DPT mat# tran# DT
MP RCT mat# tran# R C0
MP POR mat# tran#  $\Phi$ 
MP TOR mat# tran#  $\tau$ 
MP TRT mat# tran# error_tolerance_for_refinement
MP TUT mat# tran# error_tolerance_for_unrefinement
NDS string# node
FCS string# element# face#
EGS string# node1 node2
DB FLW string# Dirichlet_bc
NB FLW string# Neumann_bc
DB TRN string# tran# Dirichlet_bc
NB TRN string# tran# Neumann_bc
TC TO initial_time
TC NTS #time_steps
TC DT length_of_timestep
RUN
OC
END

```

Table A.9: ADH input file template.